IBM

S15F-8798-00

IBM 7525
Data Collection Terminal
Physical Planning and Operations Guide

First Edition (December 1988)

Use this publication only for the purpose stated in the preface.

Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or technical newsletters.

It is possible that this material may contain reference to, or information about IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country,

Publications are not stocked at the address given below. Requests for copies of IBM publications should be made to your IBM representative or the IBM branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Address any comments to IBM Corporation, Manufacturing Systems Products, Department 27X, Internal Zip 2102, P.O. Box 1328, Boca Raton, Florida 33429-1328. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Federal Communications Commission (FCC) Statement

Warning: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Instructions to User: If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna.
- Relocate the computer with respect to the receiver.
- Move the computer away from the receiver.
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

Properly shielded and grounded cables and connectors must be used for connection to peripherals in order to meet FCC emmision limits. Proper IBM cables are available from authorized dealers. IBM is not responsible for any radio or television interference caused by using other than recommended cables or by unauthorized modifications to this equipment. It is the responsibility of the user to correct such interference.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

How to Identify and Resolve Radio-TV Interference Problems.

This booklet is available from the following:

FOB Public Contact Branch
Room 725
1919 M St. NW
Washington, DC 20554
Tele. (202) 634-1940

Consumer Assistance and Small Business Division
Room 254
1919 M St. NW
Washington, DC 20554
Tele. (202) 632-7000

About This Guide

This guide provides guidance during the planning, installation, set-up, and checkout of the 7525 Data Collection Terminal.

It contains a general description of the terminal and detailed information about the following items:

- Hardware
- Typical configurations
- IBM Series/1 line connections
- IBM Personal Computer line connections
- IBM Personal System/2 line connections
- IBM Industrial Computer line connections
- Set-up parameters and download sequence
- Description of files
- Communications Control protocol
- Commands.

The appendixes are contain information regarding:

- Terminal error codes
- Configuration and ordering
- Machine readable media standards
- Physical installation
- A sample program demonstrating communication between a 7525 Terminal and a Personal Computer.

Contents

Chapter 1. Introduction

The IBM 7525 Data Collection Terminal is primarily intended for applications such as:

- Time and attendance badge reading
- Data collection
- Data inquiry.

This terminal is designed to withstand continuous use by a large number of personnel each day. Typical environments could range from retail stores to large manufacturing companies employing thousands of workers. The 7525 Data Collection Terminal includes the following features:

- A bar code wand reader and/or a slot-type card reader
- A membrane keyboard
- A one-line, 40-character, alphanumeric, liquid-crystal display (LCD).

The slot-type card reader is available as either a magnetic stripe badge/card reader or an optical bar code reader. A bar code wand connector is on all models of the 7525 Data Collection Terminal.

The terminal has a fixed standard keypad and has provision for insertion of a removable custom keypad overlay. The custom keypad overlay is a customer or user responsibility. Forms are provided in Appendix B, "Configuration Guides and Forms" to help you customize the overlay. The forms include layout of special alphabetic/function keys, logos, and colors.

The terminal can be personalized to operate in an interactive or buffered environment (store and forward). During power outages, capacitive backup is used to keep any buffered transactions stored in the terminal from being lost.

Each terminal is addressable and up to 16 terminals can be multi-dropped on a single communications line.

Chapter 2. Hardware Description

The four models of the IBM 7525 Data Collection Terminal are described on the following pages.

IBM 7525 Data Collection Terminal, Model 001

Enclosure

The enclosure is a molded plastic housing with a two-piece formed sheet metal backplate. It is designed to allow the terminal to be operated in an industrial environment. It protects the electronics from physical damage, electromagnetic interference, dust, and vibrations.

The enclosure has a medium-textured shadow-gray finish.

Membrane Keypad

A membrane keypad is mounted in the front of the enclosure. This is a combination keypad with 12 numeric (0-9, Clear and Enter) and 28 alphabetic/function keys. The operation of the alphanumeric, Enter, and Clear keys are fixed in the microcode. The function key operations are defined in a personalized customer download. The keys are on 3/4-inch centers and an audible feedback signal is provided when a key is pressed.

A keypad graphics overlay can be placed over the keypad to provide customized key captions. A slot is provided in the side of the enclosure (next to the keypad), for installing the overlay without opening the enclosure. The overlay is held in place by a screw in the back of the enclosure.

LCD Display

The 7525 Terminal has a 40-character, one-line, reflective LCD display located above the keypad. Information is shown on the display with characters 0.218 inches high. The display normally shows prompt messages, time of day (12- or 24-hour format), key entry data, bar code data, and badge data under control of the personalized customer download. The display also shows operational or system messages as directed by the initial communication setup.

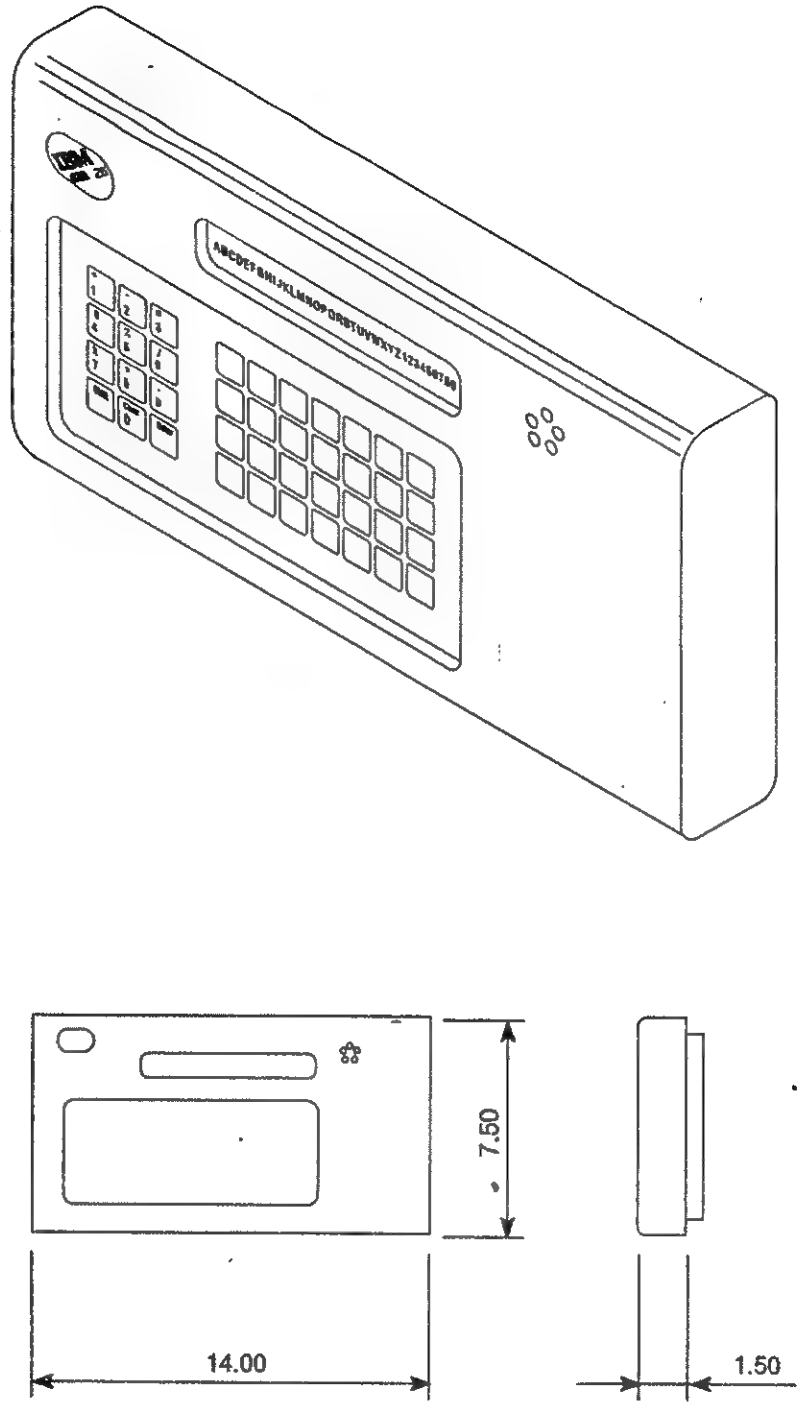


Figure 2-1. Model 001

Host Communication

Host communication is accomplished through a 4-wire RS422 interface. A polled, serial ASCII protocol allows up to 16 terminals to be multi-dropped from an IBM Series/1 RS422 port, or from an IBM Personal Computer or IBM Industrial Computer with an RS422 port installed up to a distance of 4000 (wire) feet. The customer is responsible for the configuration if any other RS422 drivers, receivers, or converters are used or if the maximum distance limitation of the communication cable is exceeded. A 9-pin, D-shell receptacle is provided for signals wired as shown in the following table.

Pin No.	Line Name	Pin No.	Line Name
1	+TX (from terminal)	6	Memory backup return
2	-TX (from terminal)	7	Ac power
3	+Rec (to terminal)	8	Ac power
4	-Rec (to terminal)	9	Ground
5	Memory backup enable		

Note: Power requirements are listed under "Power Supply" below.

Electronic Circuit Board

The electronic circuit board provides a microprocessor, memory, peripheral drivers, connectors, hardware components, and microcode necessary to provide the functions described in this manual. The circuit board is installed in a separate metal enclosure within the molded-plastic terminal enclosure.

RAM Storage

There are 32 Kbytes of RAM (random access memory) storage in the terminal, of which 7 Kbytes are used by the terminal microcode to store system variables. The other 25 Kbytes of RAM are available for user storage to hold transaction records and user downloaded files. Capacitive data-loss backup protects this storage for a minimum of 48 hours after power to the terminal is interrupted. If power is restored after 48 hours, the communication parameters have to be reset and a new personalization file (download) from the host computer is required.

ROM Storage

There are 64 Kbytes of ROM (read-only memory) in the terminal. This storage is used by the terminal microcode to provide the functional characteristics of the terminal as described in this manual.

Power Supply

The power required by the terminal is shown below. Circuitry on the electronics circuit board converts this to logic voltages. A power transformer is included with the optional desk mount accessory and wall mount accessory (listed under "IBM 7525 Data Collection Terminal Accessories" on page 2-6) to provide 16 Vac to the terminal from a 120 or 220 Vac power source.

- Ac input: 12 to 24 Vac (absolute minimum/maximum)
- Dc input: 16 to 35 Vdc (absolute minimum/maximum)

Low-Power Detect

Power sensing circuitry monitors main power and forces an orderly shutdown if a low power condition (brownout) occurs. This feature gives the terminal the ability to "warm start": the terminal restarts in the same mode it was in before the power failure. Data entered into the terminal prior to the power failure remains in the transaction buffer up to 48 hours while the power is off.

Time-of-Day (TOD) Clock

A time-of-day clock chip maintains the time and date within the terminal. It is initially set by a command from the host computer (the host can also read the time-of-day), and terminal microcode can read the TOD to "time stamp" transactions. The clock chip is accurate to within six seconds a day. Operation of the clock chip is maintained during power outages up to two hours using capacitive backup technology (no batteries). If power is restored to the terminal after two hours, the time has to be reset.

Watchdog Timer

If the 7525 terminal microcode gets into a loop or the terminal processor is not running, the watchdog timer times out and forces a hardware reset. The microcode then executes power-up diagnostics and cold-start routines.

Audible-Tone Generator

A transducer provides an audible tone under control of terminal microcode, or directly by command from the host computer. The audible tone provides operator feedback for the membrane keypad and good/bad read indications for the bar code wand and bar/magnetic slot readers.

Bar Code Wand Connector

The bar code wand reads Code 3 of 9, Interleaved 2 of 5, EAN-8, EAN-13, and UPC/A bar codes. A 9-pin, D-shell plug accepts a digital bar code optical wand male connector wired as shown below.

Pin No.	Line Name	Pin No.	Line Name
1	+ 5 Vdc	6	Unused
2	Unused	7	Unused
3	Pen data	8	Unused
4	Signal ground	9	Unused
5	Shield		

IBM 7525 Data Collection Terminal, Models 002, 003. and 004

The Model 002 includes all the features of the Model 001 plus a visible-light bar code slot reader that reads Code 3 of 9, Interleaved 2 of 5, EAN-8, EAN-13, and UPC/A bar codes.

The Model 003 includes all the features of the Model 001 plus an infrared light bar code slot reader that reads Code 3 of 9, Interleaved 2 of 5, EAN-8, EAN-13, and UPC/A bar codes.

The Model 004 includes all the features of the Model 001 plus a magnetic slot reader that reads ABA track 2, or full-width NRZ (non-return to zero) F2F encoding. Both a numeric and a 63-character, character set are supported, with a maximum record length of 99 characters.

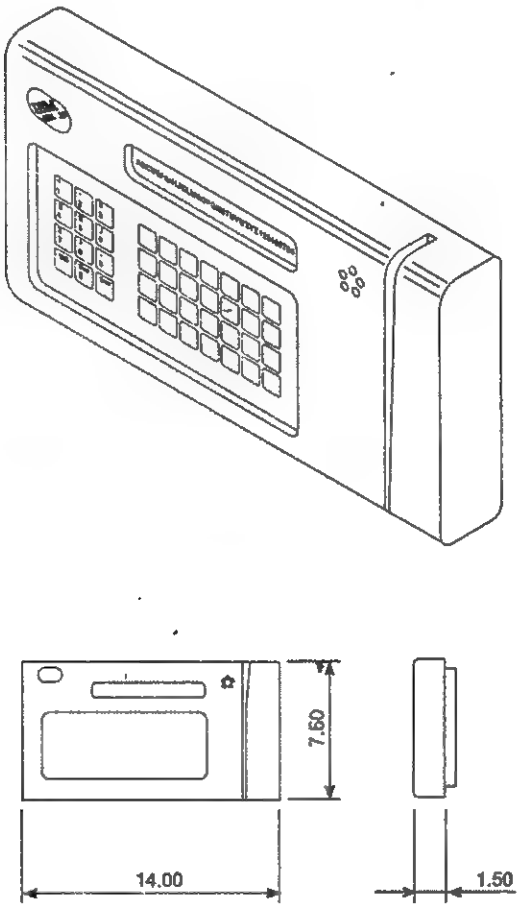


Figure 2-2. Models 002, 003, and 004

IBM 7525 Data Collection Terminal Accessories

These optional accessories are available for all models of the 7525 Data Collection Terminal:

- Wall mount accessory

  This accessory has a built-in keylock that is used to fasten the terminal securely to a mounting plate attached to a wall. The wall mount unit includes an internal power module that provides power for the terminal from a 120/240 Vac primary power source. The cable from the wall mount to the terminal and the terminal mounting plate are included.

- Desk mount accessory

  This accessory acts as a base to hold the terminal at a 14-degree angle on a desk. The desk mount also has a built-in keylock for fastening the terminal to the terminal mounting plate. The desk mount unit includes an internal power module that provides power for a terminal from a 120/240 Vac primary power source. The cable from the base to the terminal and the terminal mounting plate are included.

- Bar code wands
  - A high-resolution, 0.15 mm (0.006 in.) infrared light digital bar code wand
  - A high-resolution, 0.15 mm (0.006 in.) visible-light bar code wand
  - A low-resolution, 0.25 mm (0.010 in.) visible-light bar code wand
  - A low-resolution, 0.25 mm (0.010 in.) infrared light bar code wand.

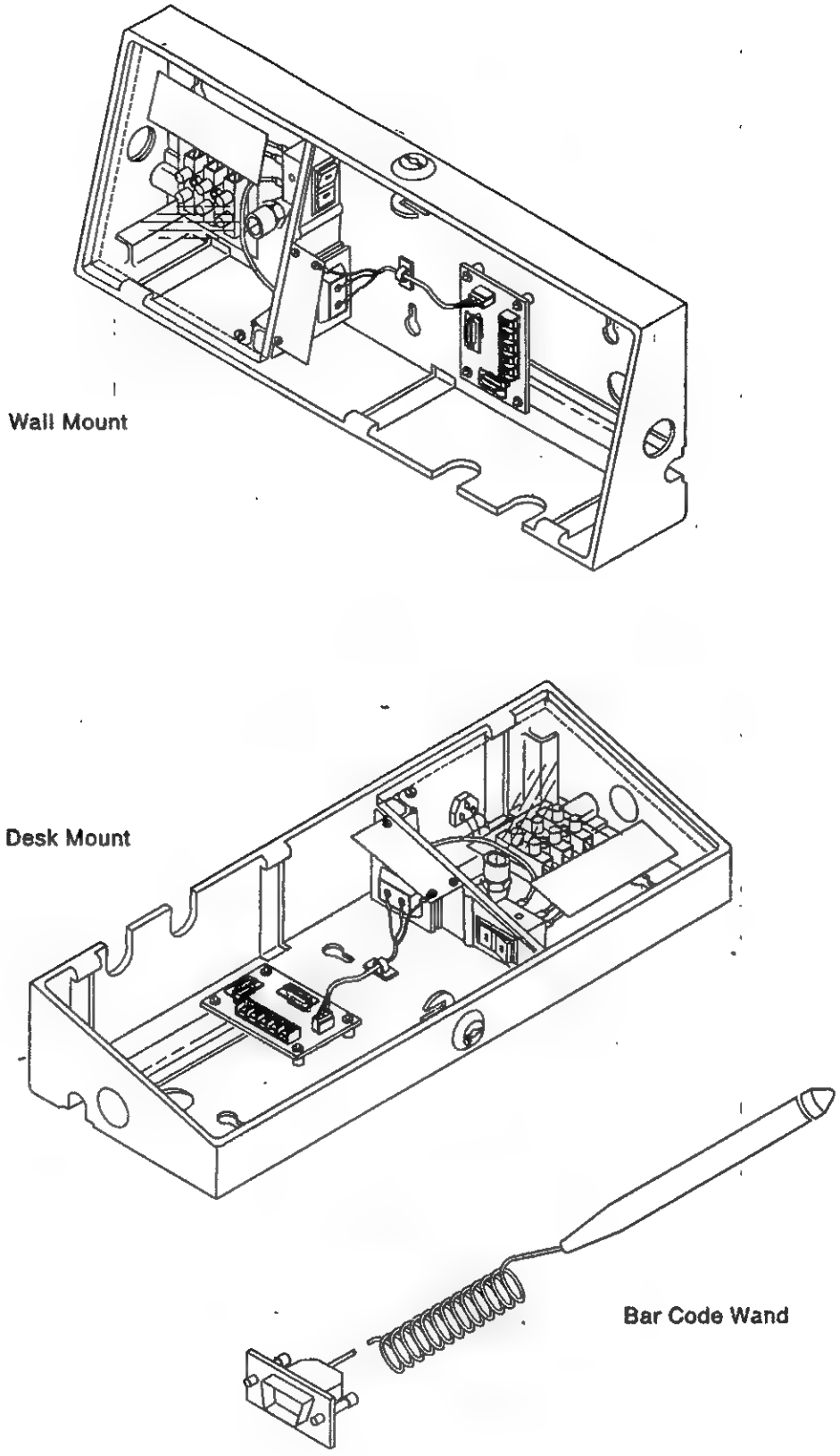


Figure 2-3. Data Collection Terminal Optional Accessories

Chapter 3. Attaching to a Host Computer

IBM Series/1 Computer

The most popular way of attaching 7525 terminals to the IBM Series/1 computer is through an RS422 port on the Multifunction Attachment (MFA) card (feature number 1310). You may use a 2095 Controller and Eight-Line Adapter Card (RPQ D02350) to attach the terminals. The MFA card is shown in "Multifunction Attachment Card Jumpers" on page 3-4.

1. Install both B1 jumpers for multi-drop RS422 lines. If multi-drop lines are not used, install both B2 jumpers.
2. Remove jumpers MT, S0, S1, and S2.
3. Jumpers D2 and D4 determine the maximum number of ports used on the card. Install the jumpers as follows:

Device Count	Jumper D2	Jumper D4
1	Install	Install
2	Install	Remove
3	Remove	Install
4	Remove	Remove

4. Jumpers L0 through L6 determine the Series/1 base device address for the card as follows:

Series/1	MSB							LSB
Address	8	9	10	11	12	13	14	15
Jumper	L0	L1	L2	L3	L4	L5	L6	0

MFA Identification Word (0011 0000 0011 0110)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Word	0	0	1	1	0	0	X	X	0	0	0	0	0	0	0	0

Refer to "Multifunction Attachment Card Jumpers" on page 3-4. Jumpers are not required for bits 0 through 5.

D = Device Count (Domain) Jumpers (Bits 6 and 7)

The relationship between the two device count jumpers and the ID Word (bits 6 and 7) is shown in the following table.

Device Count	Jumper D2	Jumper D4	ID Bit 6	ID Bit 7
1	On	On	0	1
2	On	Off	0	1
3	Off	On	1	0
4	Off	Off	1	0

L = Logical Jumpers - Series/1 Device Address (Bits 8 through 15)

The logical jumpers are related to the ID Word (bits 8 through 15) as shown in the following table. Bit 15 is always 0.

Jumper on equals logical 0.

ID bit	8	9	10	11	12	13	14	15
Jumper	L0	L1	L2	L3	L4	L5	L6	0

Bias Jumpers

The bias jumpers must be on their respective "1" pins (Mark).

MT = Multipoint Tributary (MT) Jumpers

The multipoint tributary jumper should not be installed. When the MT jumper is off, jumpers S0, S1, and S2 take on the following meaning.

S0 on	IPL disabled
S1 on	Answer tone provided
S2 on	Switched line in use

CX - Secondary Station Address Jumper

When the MT jumper is on, the secondary station address is CX, where X is determined by the use of jumpers S0, S1, and S2.

Jumper on equals logical 1.

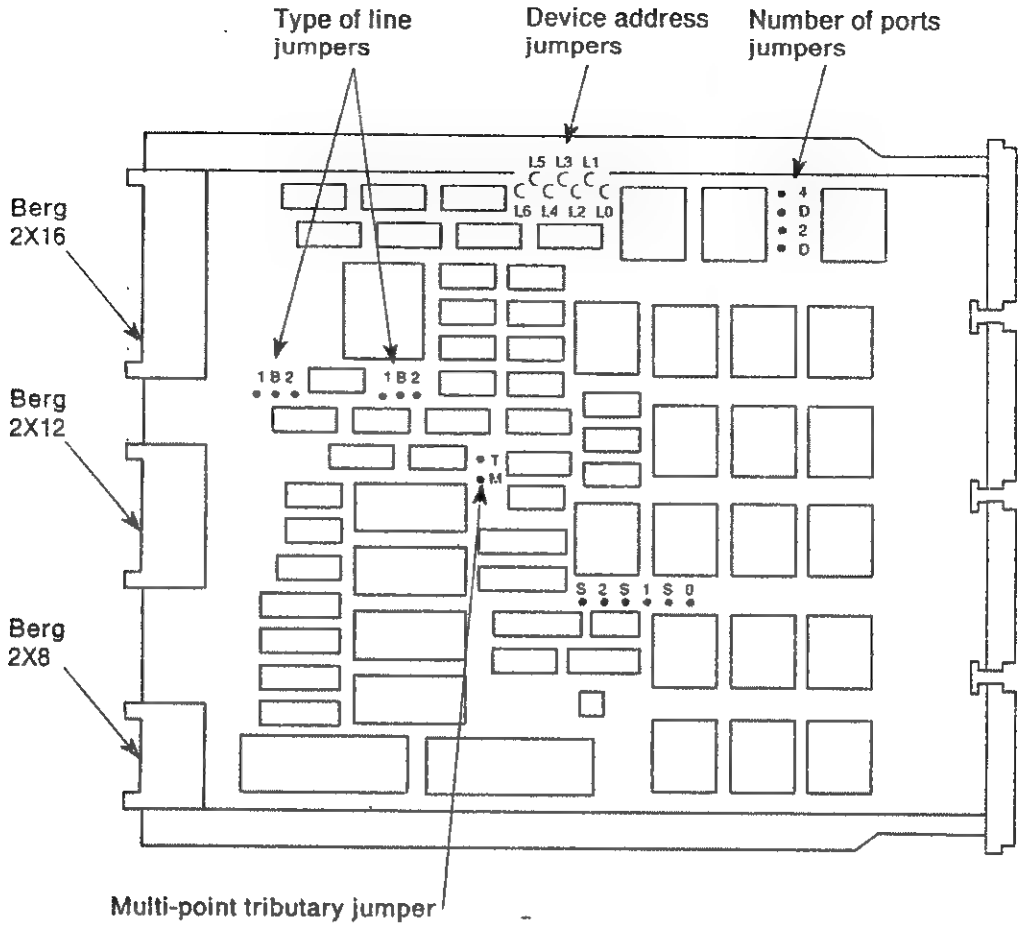
Secondary Station Address	Jumper S2	Jumper S1	Jumper S0
C0	Off	Off	Off
C1	Off	Off	On
C2	Off	On	Off
C3	Off	On	Off
C4	On	Off	Off
C5	On	Off	On
C6	On	On	Off
C7	On	On	On

Port Addresses

When the address jumpers for a card are set, they determine the base address of the card. All addresses are relative to a base address. Base address 58 is used as an example.

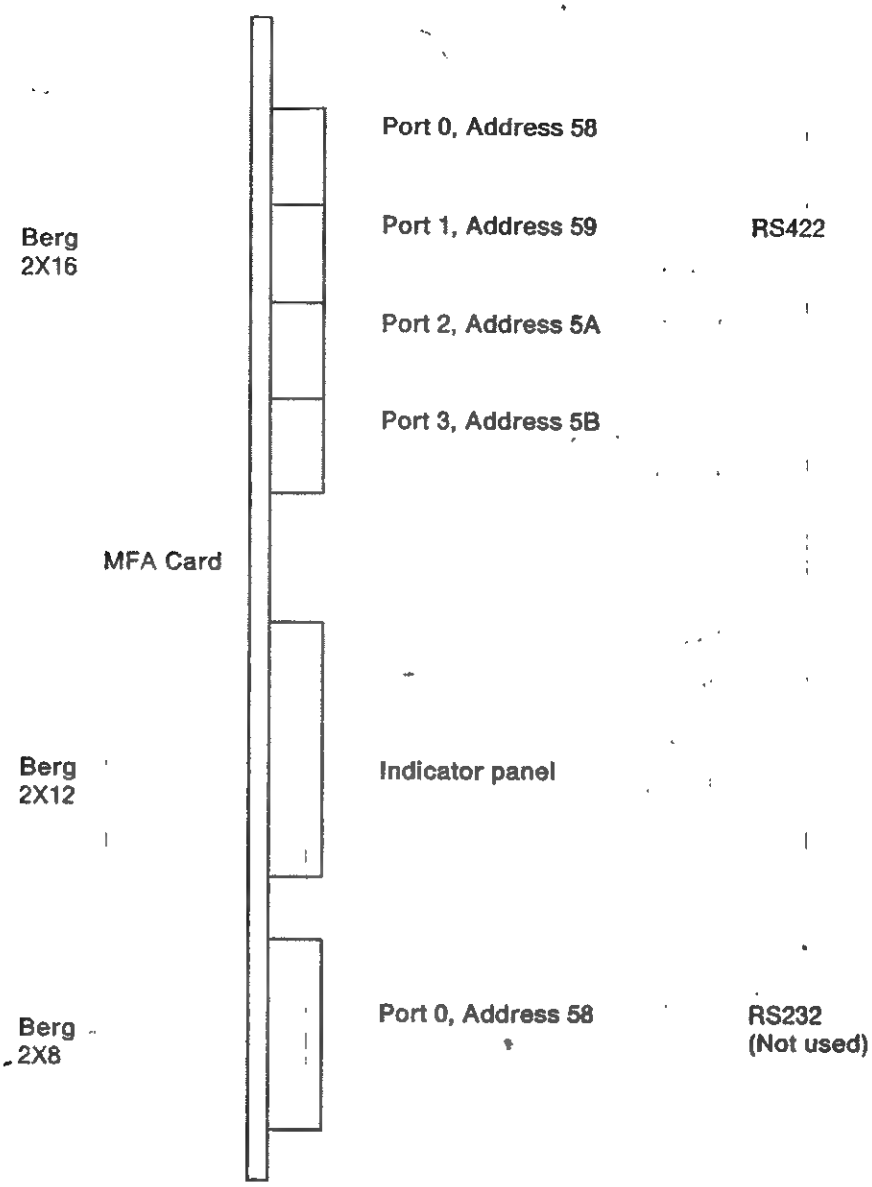
"Multifunction Attachment Card Ports" on page 3-5 shows the location of the port connectors on the Series/1 RS422 feature card.

Multifunction Attachment Card Jumpers



Multifunction Attachment Card Ports

The Multifunction Attachment Card ports 1, 2, and 3 are used for RS422. Port 0 can be used for RS422 or RS232.



IBM Personal Computer, Personal System/2, and Industrial Computer

RS422 Communications Adapter Card

An RS422 or RS422/RS485 Communications Adapter Card is used to attach 7525 Data Collection Terminals to any of the following IBM computers:

- Personal Computer AT
- Personal System/2 Model 30
- Personal System/2 Model 50
- Personal System/2 Model 60
- Personal System/2 Model 80
- 7531 Industrial Computer
- 7532 Industrial Computer.

The Communications Adapter card utilized should include the following features:

- RS422 or RS422/RS485 balanced line drivers and receivers
- Designed for multi-drop operation
- Programmable baud rates from 1200 to 9600
- Configurarable as COM1 or COM2.

Chapter 4. Operation Modes and Parameters

Modes of Operation

The 7525 Data Collection Terminal communicates with a host computer using a polled, asynchronous serial ASCII protocol. The terminal builds and time stamps transactions to be sent to a host computer. The format of the transactions and editing of input can be customized by sending the terminal a parameter file from the host computer. Changing the terminal characteristics or how it builds, edits, and time stamps transactions is as simple as sending the terminal a new parameter file from the host computer.

Interactive Mode

Interactive mode allows the terminal to build a transaction and lock out any further activity until the transaction is acted on by the host, or until a programmable (download parameter) time-out function occurs. If time-out occurs, the waiting transaction is cancelled and an out-of-service message is displayed on the terminal screen. Transactions may not be entered until the host resumes polling the terminal.

Interactive/Buffered Mode

Interactive/buffered mode allows the terminal to build a transaction and lock out any further activity until the transaction is acted on by the host, or until a programmable time-out function occurs. If time-out occurs, the waiting transaction is stored in a buffer and the terminal can accept new transactions. The time-out can be set to a value through a download parameter which gives sufficient time for the host to poll the terminal. The terminal then operates in buffered mode as described in the next paragraph until the host starts to poll the terminal again.

Buffered Mode

Buffered mode allows the terminal to place all transactions in the buffer when they are entered. The host poll retrieve the transactions in first in/first out (FIFO) order as they were entered in the buffers. The operator may continue to enter transactions until the "BUFFER FULL" message displays on the terminal screen. The host then must poll some of the transactions from the terminal before an operator can enter any new transactions.

Badge Initiated Transactions

Normally the 7525 Terminal initiates a transaction when the operator presses a function key. Certain applications may require that an operator needs only to read in a badge to initiate a transaction. These functions can be implemented using time-of-day for fast clocking and an optional badge function key definition for reading a badge first.

Fast Clocking

Fast clocking lets transactions be created with only the reading of a badge (no function key required). This can be used during a factory shift change, when only badge data is collected for time and attendance purposes. Fast clocking transactions are buffered and retrieved by the host in FIFO order from the buffer, regardless of the terminal operating mode.

Optional Badge Function Key Definition

An optional function key definition allows the reading of a badge to initiate a transaction. This badge-initiated transaction may require other input data which can be defined in the optional function key definition.

Set-up Parameters and Download Sequence

When initially powered on, the terminal is in a programming mode for selection of the following communication set-up parameters, selected by use of the numeric keypad. Once selected, these parameters are stored in RAM, and are retained there until main power is lost and capacitive data-loss backup power has been discharged. Power discharge takes a minimum of 48 hours after the main power has been removed from the unit.

Communication parameters may be changed by use of a special reprogramming badge (all zeros), or by disconnecting from main and capacitive backup power. To do this, remove the power and communication 9-pin, D-shell connector located inside the customer access panel. If you have the optional desk or wall mount, just disconnect the terminal from the mount.

Selecting Initial Communication Parameters

When the terminal is first powered on, it is in programming mode and the initial communications parameters must be selected using the numeric keypad:

1. When power is applied to the terminal for the first time, or after the terminal's power and communication connector has been disconnected, or after the 48-hour capacitive data-loss backup is discharged, "7525 VX.X COPYRIGHT IBM CORP. 1987, 1988" is displayed (VX.X is the version level of the microcode).
2. Press Enter and "ADDR-A" is displayed.
   a. Press any numeric key and a new address is displayed. There are 16 address choices, from A through P. Continue displaying address choices until you have the one you want. *Each terminal must have a unique address for proper communication on the loop.*
   b. Press Enter to set the address.
3. After the address is set, "BAUD RATE - 9600" is displayed.
   a. Press any numeric key to display a new baud rate. The choices are 1200, 2400, 4800, and 9600.
   b. Press Enter to set the desired baud rate.
4. After the baud rate is set, "DATA CHAR LENGTH-8" is displayed. There are two choices for character length, 7 and 8.
   a. Press any numeric key to display the desired character length.
   b. Press Enter to set the data character length.
5. If you selected 7, "EVEN PARITY" is displayed.
   a. Press any numeric key if you want "ODD PARITY" displayed.
   b. Press Enter to set the parity for the terminal.
6. After the data character length and parity are set, "OUT OF SERVICE DOWNLOAD NEEDED" is displayed. Initial communication parameter

selection is now complete; and the terminal is ready for a download from the host computer.

Reprogramming Badge Procedure

Communication parameters may be changed after a terminal has been downloaded if the reprogramming parameter in File 0, record 00 is set during download. A reprogramming badge containing all zero characters must be used. The number of characters is the same as specified in File 0, Record 00 for fast clocking.

To reprogram the terminal:

1. Read the all-zero badge.
2. Select parameters using the numeric keypad the same as when selecting initial communication parameters.

Download Sequence (Initialize and Load the Terminal)

A predefined download sequence is required for loading operating parameters in the terminal. This initialization sequence reserves RAM storage space for each of the appropriate files (0—9). Initial text is also loaded into the appropriate files (0—9). Once loaded, these parameters remain in capacitive backed up RAM until a new download sequence is received, or until the cable that provides power and communication to the terminal has been disconnected.

The download sequence is shown in the following table:

Step	Command	Terminal operation
1	A 0	Take terminal off-line
2	1	Check status
3	9B 0 text	Initialize File 0
4	1	Check status
5	9B 1 text	Initialize File 1
6	1	Check status
7	9B f text	Initialize Files 2 through 7 (f = file to initialize). See note.
8	1	Check status after each file is initialized (2 through 7)
9	9B 8 text	Initialize File 8
10	1	Check status
11	9B 9 text	Initialize File 9
12	1	Check status
13	9C 0 text RS	Load text to File 0. RS = record separator
14	1	Check status
15	9C 1 text RS	Load text to File 1. RS = record separator

Step	Command	Terminal operation
16	1	Check status
17	9C f text RS	Load text to Files 2 through 7 (f = file to be loaded). RS = record separator. See note.
18	1	Check status
19	9C 8 text RS	Load text to File 8. RS = record separator
20	1	Check status
21	8 text	Specify current buffer for polling
22	1	Check status
23	4 text.	Set clock
24	1	Check status
25	A 1	Put terminal on-line
26	1	Check status

The terminal is now ready to be polled by the host computer.

Note: File 9 must always be initialized. Files 0, 1, and 8 must always be initialized and loaded. Files 2 through 7 are optional; however, if any one of them is initialized, it must be loaded.

A detailed description of text and data associated with each command can be found in "Command Set" on page 6-3.

Chapter 5. Files

General Rules for Files

Download

Once a download sequence is completed, files may only be reinitialized by doing another download sequence. Individual files may not be reinitialized.

A download sequence will not be accepted by the terminal when data records are present in File 9 (transaction file). When data records are removed from File 9, download will operate correctly.

Text or Data Compaction

Text or data will be compacted (2 characters per byte) only if files are initialized as numeric. Refer to Command 9B in "CMD 9B = Initialize the Specified Text File" on page 6-10. Compaction is done internally by the terminal microcode. The following 16 characters are treated as numeric text or data and can be included in a numeric file.

0 1 2 3 4 5 6 7 8 9 : ; < = > ?

This is done to save space in the 25K user RAM and allows more records to be stored.

CAUTION:
If alphabetic text or data is loaded into a numeric file, it is compacted along with numeric text or data and the result is unpredictable.

File Initialization

When a file is initialized (command 9B), space must be allowed for an end-of-file character that will be inserted by terminal microcode. This is an ASCII carriage return character.

Validation of Keyed Data

Field length for keyed data will always be specified by a two-byte parameter (11). The range is 01 through 40. When specified, microcode will overlay the existing display message with underscores "___" to define the field. Underscores (01—40) will be placed in the right-most display positions.

- In a numeric key entry field only the numeric keys will be active and numeric characters must be entered in all positions of the field. Numeric characters enter into the far right position of the field with each new key entry, moving the previous key entry one position to the left. All characters scrolled out of the field are deleted. Press the Enter key to complete the field.
- In an alphanumeric key entry field, the space key removes underscores and advances the cursor. The fill key fills the remainder of an incomplete field with ASCII spaces ( ). Alphanumeric characters enter into the far left position of the field with each new character entering one position to the right of the previous one. Once the field is full, characters enter into the far-right position of field,

scrolling the previous characters out the left side of the field. All characters scrolled out of the field are deleted. Press the Enter key to complete the field.

Validation of key-entered data is specified by use of a two-byte text validation file parameter in File 0, Function Key Parameters. See "Function Key Parameter Descriptions" on page 5-9.

The text validation file parameter has the following meaning:

- The first byte indicates the file number "f" (2—7). If this value is 0, it means "Do not check data - Accept it as received."
- The second byte, "t," specifies that the file contains positive text (1) or negative text (0). Validation of a positive text file requires that a matching record exist in the file for data which has been entered. Validation of a negative text file requires that a matching record is not found in the file for data entered.
- When the first byte is 0, the second byte must also be 0.

Validation of Keyed Data Using Advanced Features

Maximum field length for keyed data will always be specified by a two-byte parameter (11). The range is 01 through 40. The autofill, cursor positioning, and display buffer (advanced features) optional parameters on a (:2) or (:3) function key definition must all be configured if any option is used.

- A numeric key entry field with autofill parameter (a) = 1 bypasses the length check. The Enter key may be used at any time to complete the field. Any remaining underscores in the numeric key entry field are changed to ASCII zeroes (0) and the field completed. Enter the numeric data as described in "Validation of Keyed Data" on page 5-1.
- An alphanumeric key entry field with autofill parameter (a) = 1 bypasses the length check. The Enter key may be used at any time to complete the field. Any remaining underscores in the alphanumeric key entry field are changed to ASCII spaces ( ) and the field completed. Enter the alphanumeric data as described in "Validation of Keyed Data" on page 5-1.

Field positioning parameters (pp) are two numeric characters that position the leftmost character of the field at display positions (01—40). The display is numbered from left to right.

Write-to-display buffer parameter (b) can be a 0 or 1. If you use any of the advanced features on a (:2) or (:3) function key definition you must specify all four bytes.

Validation of keyed data using the advanced features is the same as described for "Validation of Keyed Data" on page 5-1.

CAUTION:
Refer to "Advanced Features" on page 7-2 and "Function Key Parameter Descriptions" on page 5-9 for key definitions (:2) and (:3) before using the advanced features.

Validation of Badge Data

Field length for badge data will always be specified by a two-byte parameter. The range is 01 through 40. A variable length badge may be specified but the length parameter must be set to the maximum length of any badge that is to be read.

Badge data is normally not displayed; however, if the display option is specified, the microcode will overlay the existing display message with underscores "___" to define a badge entry field. Underscores (01–40) are placed in the right-most display positions. Data fills the defined field from left to right as entered. The remainder of the field is filled with ASCII space characters ( ).

Validation of badge data is specified by use of a two-byte text validation file parameter in File 0 (Function Key Parameters). Refer to "Function Key Parameter Descriptions" on page 5-9.

This parameter has the following meaning:

- The first byte shows the file number "f" (2 through 7). A 0 entry means: "Do not check data - Accept it as received."
- The second byte, "t," specifies that the file contains positive text (1) or negative text (0). Validation of a positive text file requires that a matching record exist in the file for data which has been entered. Validation of a negative text file requires that a matching record is not found in the file for data entered.
- When the first byte is 0, the second byte must also be 0.

Examples of Fixed Display Messages

The following table shows typical predefined messages the user will observe during operation of the terminal.

Display Message	When Used
OUT OF SERVICE - DOWNLOAD NEEDED	For download after power-on
OUT OF SERVICE ---- TIME -----	When off-line command is completed after download
FULL ---- TIME -----	When transaction buffer is full
CLOCK SET TO MM DD YY HHMMSS	Clock set message
7525 VX.X COPYRIGHT IBM CORP. 1987, 1988	On power-up, before initial setup. X.X is the version level of the microcode.
SET DATE AND TIME ----------	While in set-clock routine

Description of RAM Files

This section describes the use of the 10 unique files stored in the 7525 Data Collection Terminal 25K user RAM memory. Record formats for each file are also described.

The file types shown in Table 5-1 use the following definitions. These definitions or rules must be followed when personalizing a terminal for a data collection application.

Type	Description of Text or Data
P	Privileged file (text may be altered only by download)
N	Numeric file (compacted; may delete or add records, except when privileged or polled file)
A	Alphanumeric (non-compacted; may delete or add records, except when privileged or polled file)
PP	Polled file (loaded by terminal microcode; may not delete or add records)
NP	Non-polled file (may be altered by host, except when privileged)

General File Description

Table 5-1. File 0–9 General Description

File	File Name	Purpose	Records	Type
0	Operating parameters	System parameters	00	P, N, NP
		Function key parameters	01–29	P, N, NP
1	Fast clocking times	Fast clocking parameters	User defined	P, N, NP
2	User text file	Input data validation	User defined	A or N, NP
3	User text file	Input data validation	User defined	A or N, NP
4	User text file	Input data validation	User defined	A or N, NP
5	User text file	Input data validation	User defined	A or N, NP
6	User text file	Input data validation	User defined	A or N, NP
7	User text file	Input data validation	User defined	A or N, NP
8	Prompt file	Prompt messages	User defined	P, A, NP
9	Transaction file	Transaction buffer	To end of storage	P, A or N, PP

File 0 — Basic Terminal Operating Parameters

File 0 will always have 29 records (00 thru 28) each containing parameter entries that determine basic terminal operating characteristics and function key definitions. It is a numeric, privileged, non-polled file which lets the customer customize the functions for an application. Beginning with File 0, record 1, variable length records are used which are assigned to function key locations.

An optional record 29 can be added to File 0 to further define a badge initiated transaction.

Record	Characteristic or Function Defined
00	Terminal characteristics (22 bytes)
01—28	Definitions for function keys 01 through 28 (variable length up to 128 byte records)
29	Optional definition for a badge initiated transaction (variable length up to 128 bytes)

For each unused function key position, a record (parameter :9) must be transmitted to the 7525 Data Collection Terminal. All 28 function key positions must be defined even when function keys are not used. Function key definition 29 used with badge initiated transactions is optional.

The following tables describe the contents and options for records in File 0, and show examples of function key definitions.

File 0, Record 00 (Bytes 0—5) Description

Table 5-2. Bytes 0—5

Byte	Parameter Function	Parameter Entry
0	Type of keypad	0 = Calculator format 1 = Telephone format
1	Type of badge reader	0 = Magnetic stripe 1 = 3 of 9 bar code 2 = I 2 of 5 bar code 3 = UPC/A bar code 4 = Magnetic and 3 of 9 5 = Magnetic and I 2 of 5 6 = Magnetic and UPC/A
2	Buffered or interactive	0 = Buffered mode 1 = Interactive mode
3	Format of time display	0 = 12 hour (HH-MM AM/PM) 1 = 24 hour (HH-MM) 2 = 12 hour (HH-t AM/PM); t = 10ths of an hour 3 = 12 hour (HH-h AM/PM); h = 100ths of an hour 4 = 24 hour (HH-t); t = 10ths of an hour 5 = 24 hour (HH-h); h = 100ths of an hour
4	Idle display format (prompt message from File 8 is left-justified)	Time display is right-justified: 0 = Time only 1 = Time and date 2 = Date only
5	Automatic hourly time stamp transaction	0 = No 1 = Yes To prevent conflict between an input transaction and the hourly time stamp record, the transaction in progress will be completed before the automatic time stamp record is created. Because of this priority, the record may contain data which is several seconds past the hour.

File 0, Record 00 (Bytes 6—15) Description

Table 5-3. Bytes 6–15

Byte	Parameter Function	Parameter Entry
6	Time stamp for transaction records	0=HHMM 1=HHMMSS 2=YYMMDDHHMM 3=YYMMDDHHMMSS
7	Error logging mode	0=None 1=Wrong length badge 2=Unable to read and wrong length badge
8	Number of beeps for a good transaction	X=Number of "good" beeps (0—9)
9	Duration of beep for a good transaction	X=Number of quarter-second intervals for a "good" beep (0—9)
10	Number of beeps for a bad transaction	X=Number of "bad" beeps (0—9)
11	Duration of beep for a bad transaction	X=Number of quarter-second intervals for a "bad" beep (0—9)
12	File number for validation of fast clocking badges	0=No validation wanted 2–7=File number for validation
13	Type, fast clock file validation	0=Data not in file (negative) 1=Data in file (positive)
14/15	Fast clock and reprogramming badge length	XX=Length for validation (00—99) If the reprogramming badge is accepted (byte 22=1), the badge data length must match the length specified in bytes 14 and 15.

File 0, Record 00 (Bytes 16—22) Description

Table 5-4. Bytes 16—22

Byte	Parameter Function	Parameter Entry
16/17	Timer A duration	XX=Time in half-second increments (01—99) This is the host communication time out ("0" is not a valid entry).
18	Timer B duration	X=Time in half-second increments (0—9) This is the time that certain status or error messages remain on the display.
19/20	Timer C duration	XX=Time in half-second increments (00—99) This is the maximum time between operator action. For example, reading a badge and pressing a key, when in a transaction sequence.
21	Add sequence no. 00—99 to transaction record	0=No 1=Yes
22	Accept reprogramming badge.	0=No 1=Yes If the reprogramming badge is accepted (byte 22=1), the badge data length must match the length specified in bytes 14 and 15.

Function Key Parameter Descriptions

These parameters determine the prompts and operator input editing the terminal provides in building a transaction to be sent to a host computer. Record 1 corresponds to function key 1, and so on. Record 29 is optional; it allows additional information to be included in a badge initiated transaction. These records can be of variable length up to 128 bytes. This is a numeric-only file and accepts only the special characters described in the section on "Text or Data Compaction" on page 5-1. All unused keys must be defined with a (:9) key parameter.

Byte 0 of each function key record always contains the special function character (:) or (;). Byte 1 is the function identifier. Bytes 2 to 127 of function key records are used for specifying options and additional function key parameters. See Table 5-2 on page 5-6 for examples of key formats.

Table 5-5 (Page 1 of 2). Function Key Parameter Descriptions

Function Key Parameter Entries	Parameter Description
:0 rr Display prompt message on display left-justified. MSG	**rr** Specifies record number in File 8 to be displayed (00—99).
:1 llft Fixed length badge read expected. FB	**ll** Specifies length of badge data (01—40 characters). **f** Validation file number 2—7 Valid numbers. 0 = NO validation. **t** Validation type. 0 = Data not in file (negative) 1 = Data in file (positive).
:2 llft Numeric keypad input expected. Optional advanced feature: **appb** Using autofill, cursor positioning, and writing to the display buffer. NK	**llft** Same as above. **a** Autofill (optional) 0 = No fill characters, underscore to length. 1 = Underscore to length, fill data sent to host is ASCII zeroes (0). **pp** Position the left-most character of the field at display position (01—40). **b** Display buffer. 0 = *Do not* use buffer 1 = Use display buffer.
:3 llft Alphanumeric keypad input expected. Optional advanced feature: **appb** Same as above AK	**llft** Same as above. **a** Autofill (optional) 0 = No fill characters, underscore to length. 1 = Underscore to length, fill data sent to host is ASCII spaces ( ). **pp** Same as above. **b** Same as above.

Table 5-5 (Page 2 of 2). Function Key Parameter Descriptions

Function Key Parameter Entries		Parameter Description
:4 llft	Alphanumeric keys or badge input expected.	llft Same as above.
:5	Set TOD clock time and date	Use to change TOD clock manually. Can be used only in last position of a function key definition.
:8	Link function	A second function key will always follow. Must be used in the first position of a function key definition.
:9	Key not used	Must be used to define all unused key positions.
;0	"Hot" key	Initiates a transaction with key ID and time stamp (must be used at start of key definition).
;1 lld0	Read variable length badges. Use only with magnetic stripe or Code 3 of 9. *Do not* use with UPC/A or I 2 of 5 bar codes.	ll Maximum length of badge data (01—40). Terminal will left-justify data padded with ASCII spaces. d Controls for data display: 0 = *Do not* display data. No enter key required. 1 = Display data. Enter key required to complete transaction.
;2	Latching function (must be used at start of key definition).	Use to latch terminal into continuous use in mode defined for key. Remains in latched mode until the Clear key is pressed.
;5 ssll	Read the display buffer (see advanced features)	Display data can be entered as part of the transaction ss = Starting display position 01—40 ll = Length of data 01—40
;6	*Do not* transmit. (must be used at start of key definition)	No transaction created. Normally, this function is used to update the display buffer (advanced features).

Examples of Function Key Record Formats for File 0

The following shows examples of typical function key record formats for File 0, records 1--29.

Record	Record Format	Result
01	:9RS	:9 = Unused key (position A4)
02	:012:10641RS →	RS = End of record
	→	:10641 = Read badge. Validate length = 6. Validate data in File 4 with positive file search.
	→	:012 = Write record 12 (File 8) to display.
	Transaction record follows:	
	Key ID (position A5)	= 01
	Badge data	= BBBBBB
	Time stamp	= HHMMSS (See Note)
03	:015:10570:018:20300RS →	RS = End of record
	→	:20300 = Accept keyed numeric data. Validate length=3. Do not validate data.
	→	:018 = Write record 18, (File 8) to display.
	→	:10570 = Read badge. Validate length = 5. Valid data not in File 7 with negative file search.
	→	:015 = Write record 15 (File 8) to display.
	Transaction record follows:	
	Key ID (position A6)	= 02
	Badge data	= BBBBB
	Keyed data	= KKK
	Time stamp	= HHMMSS

Note: Time stamp is selected in byte 6 of File 0, record 00. See Table 5-3 on page 5-7.

Figure 5-1 (Part 1 of 3). Function Key Record Formats for File 0

Record	Record Format	Result

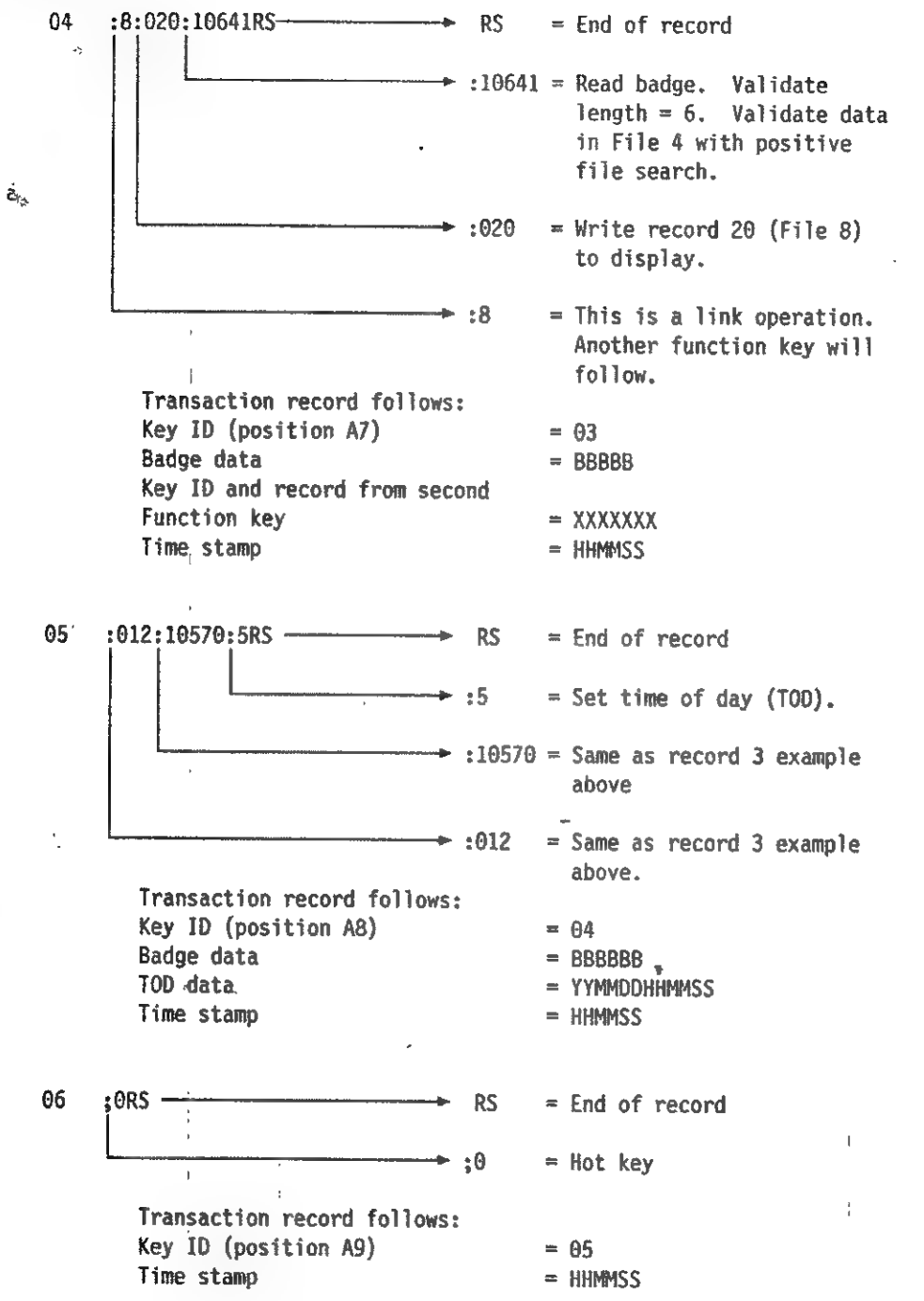


Figure 5-1 (Part 2 of 3). Function Key Record Formats for File 0

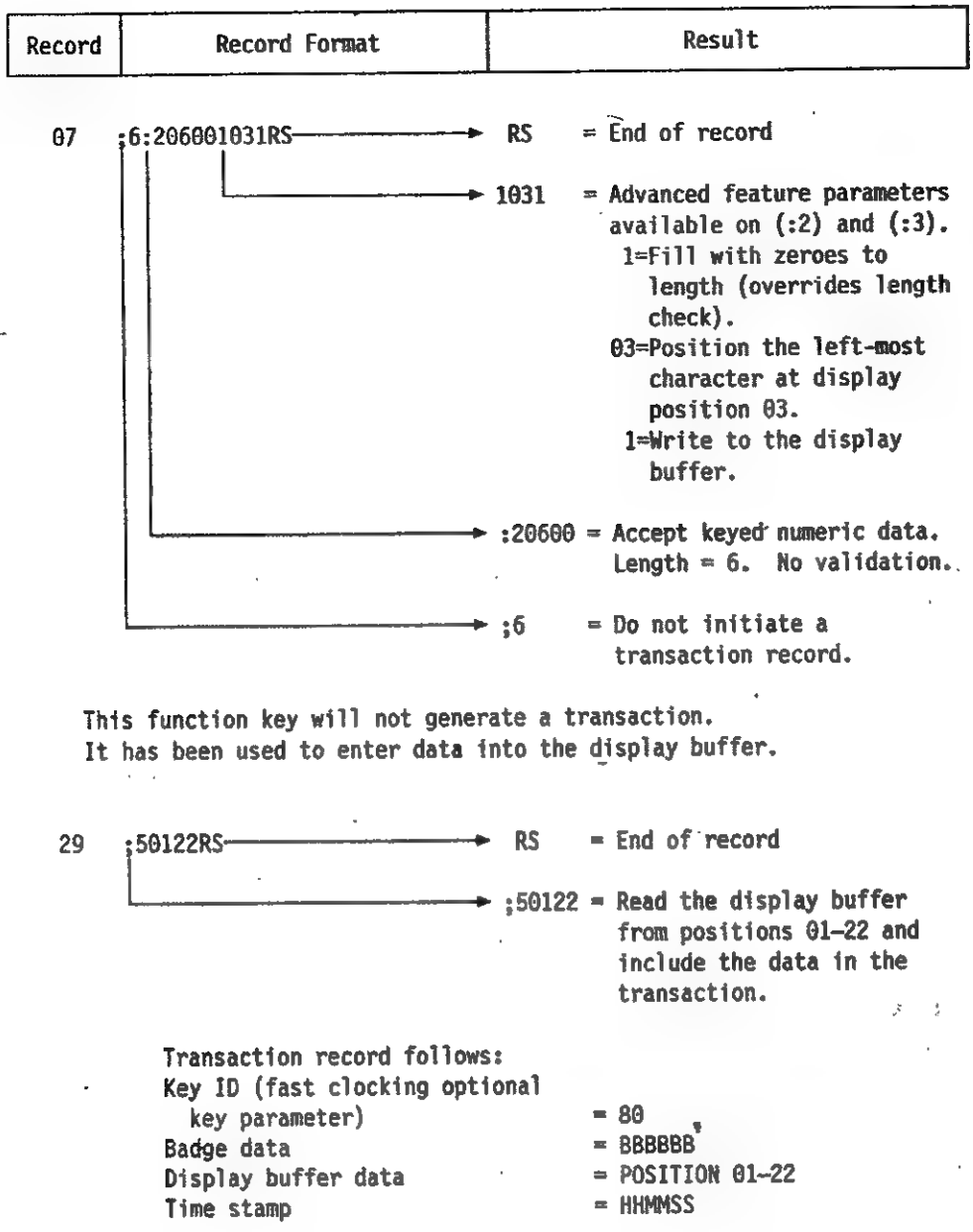


Figure 5-1 (Part 3 of 3). Function Key Record Formats for File 0

File 1 - Fast Clocking File Description

File 1 is used to hold the start and stop times when the terminal automatically switches to a fast clocking time period. During these times the terminal allows the reading of a badge to initiate a timestamped transaction of the badge read to be put into the transaction buffer (File 9).

Record	Function	Parameter Entries
00	Fast clocking time periods	:=Fast clocking not used, *or* ;=Continuous fast clocking, *or* HHMM:HHMM (24 hour format; 0000=midnight) Start:Stop
01—99	Additional time periods	HHMM:HHMM Same as above Start:Stop

Note: When fast clocking periods are used, start and stop time records are loaded into File 1 beginning with record 00. When fast clocking is not used a (:) must be loaded into record 0 File 1. Records are loaded in a sequential manner so that terminal microcode can use them in the sequence in which they are loaded. The number of records is user defined. The following loading format is used. A record separator RS must end each record entered, and a (:) must separate start time from stop time.

The format is:

START-TIME:STOP-TIME RS START-TIME:STOP-TIME RS

An example entry would be:

```
0700:0830RS1600:1700RS
```

Files 2 through 7 - Text Verification Files Description

Text verification files are numeric or alphanumeric, non-privileged, non-polled files. They may contain variable length text records for validation of data entered from the keypad or from a badge reader.

The number of records is user defined based on the amount of storage allocated for these files.

File 8 - Prompt Message File Description

This is an alphanumeric, privileged, non-polled file. It contains prompt message records that are written to the display under control of terminal microcode, and the function key parameter (:0). Prompt message records may be of variable length up to 40 characters. These records are left-justified (first character placed in position 1 of display). Any offset from the far left position (position 1) will require that blanks be put into the record or use an escape sequence.

The total number of records is user defined however, the first 10 records (00—09) are reserved prompt messages which must always be downloaded by the user. These are reserved for the functions indicated in the following figure. Actual wording of messages is user defined. A record separator (RS) must end each record. A message record less than 40 characters will terminate with a record separator (RS), and the right-most positions of the display will be blank through position 40.

Record	Prompt Message Description
00	Normal idle loop prompt
01	Timeout prompt
02	Invalid length prompt
03	Invalid data prompt
04	Invalid operation prompt
05	Fast clocking idle loop prompt
06	Interactive mode prompt (waiting for host)
07	Interactive mode prompt (host down, buffering data)
08	Good prompt (data collection sequence completed correctly)
09	Link prompt
10—99	Optional depending on the application

File 9 - Transaction Buffer File Description

File 9 is a polled file (transaction buffer). It is used by terminal microcode for storage of buffered transaction records. Records are retrieved in a FIFO sequence by polling from the host.

All transaction records begin with a function ID as shown in Table 5-6. Transaction data follows in the sequence as it was entered. The last field of each transaction record is the time stamp. If the "forced" time transaction option is selected, where File 0, record 00, byte 5=1 (see Table 5-2 on page 5-6), a record containing date and time will be created on each hour. The format of the transaction (from byte 2 on) follows the key format definition made by the user in File 0.

Table 5-6 (Page 1 of 2). File 9 – Transaction Records

Parameter Entries: Bytes 0 and 1 (See note.)	**Added Fields: Bytes 2—127**
00=Function key A4	Transaction data and time stamp.
01=Function key A5	Same as above.
02=Function key A6	Same as above.
03=Function key A7	Same as above.
04=Function key A8	Same as above.
05=Function key A9	Same as above.
06=Function key A10	Same as above.
07=Function key B4	Same as above.
08=Function key B5	Same as above.
09=Function key B6	Same as above.
10=Function key B7	Same as above.
11=Function key B8	Same as above.
12=Function key B9	Same as above.
13=Function key B10	Same as above.
14=Function key C4	Same as above.
15=Function key C5	Same as above.
16=Function key C6	Same as above.
17=Function key C7	Same as above.
18=Function key C8	Same as above.
19=Function key C9	Same as above.
20=Function key C10	Same as above.
21=Function key D4	Same as above.

Table 5-6 (Page 2 of 2). File 9 – Transaction Records	
Parameter Entries: Bytes 0 and 1 (See note.)	**Added Fields: Bytes 2–127**
22 = Function key D5	Same as above.
23 = Function key D6	Same as above.
24 = Function key D7	Same as above.
25 = Function key D8	Same as above.
26 = Function key D9	Same as above.
27 = Function key D10	Same as above.
28 = Date and time	Hourly record contains date and time
29 = Fast clocking record follows	Badge data and time stamp
30 = Badge initiated record follows	Contains badge data, transaction data, and time stamp
4XX = Error log record follows	XX = Function key used when error was detected while editing data
9XX = Error log record follows	XX = Error detected while reading magnetic badges

Note: In interactive mode, the value 50 will be added to the function ID (bytes 0 and 1). For example, key A4 becomes 50, key A5 becomes 51, and so on. If data is buffered because of host time out, the value 50 will not be added to the function ID.

Chapter 6. Commands

The command descriptions in this chapter use commas, spaces, legends, and symbols to clarify the examples for the reader. They are not sent as part of the text or data stream that is transmitted or received by the terminal.

The following legends and symbols are used in describing the commands.

Legend	Description
=>	Transmission from host to terminal.
<=	Transmission from terminal to host.
addr	Terminal address (Lower case ASCII "a" through "p," or "z" for broadcast).
CMD	Command ID character sent to a terminal.
CS1, CS2	Checksum characters.
data	ASCII data characters sent to host from a terminal as a result of a poll or command.
text	Parameters and data associated with a command sent to a terminal from the host.
RS	Record separator.
f	File identifier (used in commands 9A through 9F, B, and C).

Communication Operations

Poll

The host requests a terminal to transmit the oldest entry from the current polled File 9 (transaction buffer) or the interactive buffer. The number of records to be sent is defined by the last Command 8 received at the terminal.

Each poll returns a block of data records (the number of records is set by the previous Command 8) with a maximum length of 128 bytes. Only complete records are transmitted (records are not split). The oldest records in the file are transmitted first. Once records are transmitted, they remain in the file until the terminal receives an ACK response from the host computer. On receipt of ACK, records are flushed sequentially out of the file. The next poll then acts on the follow-on records to be transmitted. This continues until the file is emptied. When all files are transmitted, the terminal responds to a poll with NAK.

Note: A maximum of 128 bytes may be transmitted in response to a poll. Since a record cannot be split between two polls, actual data may be less than 128 bytes.

```
From host:                  => STX, addr, addr, ETX

Terminal response:
            If no data      <= NAK

            If data         <= STX, data, CS1, CS2, ETX

            If error        <= NAK

  If download is needed     <= DC1

Host reply:
         If receive ok      => ACK (release data buffer)

            If error        => NAK (hold data for next poll)
```

Command

The following shows a command or message from the host to *one* terminal with a response from the terminal to the host.

```
From host:                  => STX, addr, addr, CMD, text, RS, CS1, CS2, ETX


Terminal response:
          If receive ok     <= ACK

             If error       <= NAK

             If data        <= STX, text, CS1, CS2, ETX


Host reply:
          If receive ok     => ACK

             If error       => NAK
```

Broadcast

The following shows the host broadcasting a command to *all* terminals on a multi-dropped host port with a response from terminal "a."

```
From host:                  => STX, z, z, CMD, text, CS1, CS2, ETX

Master terminal response: <= ACK or NAK
(address "a" only)
```

Broadcast may be used only with commands 0, 2, 4, 5, 8, 9B, 9C, 9D, 9E, A, B, C and H.

Checksum

Messages transmitted and received by a terminal must have two ASCII checksum characters (CS1, CS2) just ahead of the ETX character, as shown in the preceding examples. Exceptions to this rule are:

- The poll message from the host
- All ACK and NAK responses
- The DC1 response from the terminal when a download is required.

Terminal microcode always calculates and transmits two checksum characters (CS1, CS2). Host system programming may choose to strip off or ignore them when received.

When transmitting to the terminal, the host system may elect to replace calculated checksum characters with two ASCII space ( ) characters. Space characters received at the 7525 Data Collection Terminal bypass the checksum calculation for that transmission.

Checksum characters (CS1, CS2) are calculated as follows.

- From Terminal

  Starting with the first character following the STX character, each byte is added (dropping the carry). The resulting value is converted to two ASCII characters (0—F) and transmitted just ahead of the ETX character.

- From Host

  Starting with the first character following the second address character (addr), each byte is added (dropping the carry). The resulting value is converted to two ASCII characters (Ø,F) and transmitted just ahead of the ETX character.

Command Set

CMD 0 = Reset Terminal and Execute Diagnostic Sequence

```
=> STX, addr, addr, 0, CS1, CS2, ETX
```

No terminal response. First status character is set to 1, and terminal is operational five seconds after receiving this command.

CMD 1 = Transmit Terminal Status Character to Host

```
=> STX, addr, addr, 1, CS1, CS2, ETX

<= STX, data, CS1, CS2, ETX

=> ACK or NAK
```

The ACK and NAK responses from the terminal indicate that the transmission was received with (or without) correct parity and checksum. If the terminal cannot execute a command, an error status message specifies the reason. The error status message is returned to the host in response to a Command 1.

Error Status Message Format

The *data* portion consists of a string of 12 ASCII characters that define the terminal status as shown in the table:

Position	Character	Terminal Status
1st	0	Normal state. (See following explanation for 1 state.)
	1	This is the first status inquiry since terminal power-up or reset diagnostics was completed. It resets to 0 upon receipt of ACK from the host.
2nd	0	Last command executed correctly
	1	Last command did not execute correctly
	2	Last broadcast was received correctly
3rd	0	Required file download is present
	1	File download required
4th	0	Terminal is on-line (ready)
	1	Terminal is off-line (not ready)
5th	0	Auxiliary ASCII buffer is ready to receive data
	1	Auxiliary ASCII buffer is busy
6th and 7th	cc	Identifies command for status information. Character 11 is a blank (ASCII space) for single character commands.
8th and 9th	00	Last command OK
	0f	Last command OK (f = file 0–9)
	10	Invalid command
	1f	Invalid command (f = file 0–9)
	20	Invalid text or format
	2f	Invalid text or format (f = file 0–9)
	3f	File overrun (f = file 0–9)
	4f	File not initialized (f = 0–9)
	5f	Record not found (f = 0–9)
	6f	Invalid attempt to initialize File "f" when File 9 contains data records
	69	Invalid attempt to initialize File 9 when it contains data records.
10th and 11th	ff	Identifies the current file for polled data as specified by the last Command 8 received. (See "CMD 8 = Specify Current Polled Buffer or File" on page 6-7.) Character 7 is a blank (ASCII space) for single character commands.
12th	0	No data is in transaction buffer
	1	Data is in the transaction buffer

CMD 2 = Write to Terminal Display

```
=> STX, addr, addr, 2, text, CS1, CS2, ETX

<= ACK or NAK
```

The *text* portion consists of ASCII characters to be displayed in the sequence sent. Space characters show as blanks on the display. Normally a write to the display will time out (according to the timer B count as specified in the download of File 0).

There is a 40-character display buffer in the 7525 Data Collection Terminal that can be written to and the information does not time out; it remains on the display. To write to the display buffer, use an ASCII device control 2 (DC2) character as the first character of the text.

To prevent disturbing normal transaction display messages, this command is handled only during terminal idle cycles.

CMD 3 = Transmit Terminal Microcode Level to Host

```
Format:
    =>  STX, addr, addr, 3, CS1, CS2, ETX
    <=  STX, data, CS1, CS2, ETX
    =>  ACK or NAK
```

The data portion consists of a string of two ASCII characters defining the terminal microcode where:

```
1st Char   =  1    ( Microcode Version Number )
2nd Char   =  0    ( Microcode Revision Number )
```

CMD 4 = Set Terminal Time of Day (TOD) and Calendar Date

```
=> STX, addr, addr, 4, text, CS1, CS2, ETX

<= ACK or NAK
```

The *text* portion consists of 12 ASCII numeric characters in the following format with 24-hour clock time:

```
YYMMDDHHMMSS
```

CAUTION:
When setting the TOD clock from a host computer, the 7525 Data Collection Terminal must first be made *Not Ready*. This prevents conflict between the set clock command and an input transaction which may be in progress.

The correct sequence is:

1. Set terminal Not Ready (CMD A)
2. Set TOD clock
3. Set terminal Ready (CMD A)

A transaction in progress is aborted when the unit is taken off-line. The operator has to restart the transaction.

Setting the TOD Clock

The format for setting the TOD clock either from a host (Command 4) or from the terminal keypad is YYMMDDHHMMSS. The ranges of valid parameters are listed below. Values outside these ranges will be rejected by the terminal. Status bits indicate any invalid text or format.

Month (MM)		Range of Valid Days (DD)
01	(Jan)	1 through 31
02	(Feb)	1 through 29 (for leap year)
03	(Mar)	1 through 31
04	(Apr)	1 through 30
05	(May)	1 through 31
06	(Jun)	1 through 30
07	(Jul)	1 through 31
08	(Aug)	1 through 31
09	(Sep)	1 through 30
10	(Oct)	1 through 31
11	(Nov)	1 through 30
12	(Dec)	1 through 31

Valid ranges for Time of Day (TOD) 24-hour format are:

Hours (HH)	00 through 23
Minutes (MM)	00 through 59
Seconds (SS)	00 through 59

Note: Either 0000 or 2400 is acceptable for midnight. After midnight only the 000X format is acceptable.

CMD 5 = Turn-on Terminal Audible Tone

```
=> STX, addr, addr, 5, text, RS, CS1, CS2, ETX

<= ACK or NAK
```

The *text* portion consists of a tone duration ASCII numeric character (1—9). That value multiplied by 1/2 determines the duration of the tone pulse in seconds.

To prevent disturbing normal transactions, the use of this command is handled only during terminal idle cycles.

CMD 6 = Transmit Terminal Feature Characters to Host

```
=> STX, addr, addr, 6, CS1, CS2, ETX

<= STX, data, CS1, CS2, ETX

=> ACK or NAK
```

The *data* portion consists of a string of four ASCII characters defining features of the terminal:

Position	Character	Terminal Feature Definition
1st	5	40-character alphanumeric display size
2nd	1	No RAM storage
	4	25-Kbyte RAM storage
3rd	0	TOD clock not installed
	1	TOD clock installed
4th	0	Auxiliary ASCII port not installed
	1	Auxiliary ASCII port installed

CMD 7 = Transmit Terminal TOD Clock to Host

```
=> STX, addr, addr, 7, CS1, CS2, ETX

<= STX, data, CS1, CS2, ETX

=> ACK or NAK
```

The *data* portion of the terminal response consists of the current terminal date and time using the same format as Command 4 (YYMMDDHHMMSS).

CMD 8 = Specify Current Polled Buffer or File

```
=> STX, addr, addr, 8, text, RS, CS1, CS2, ETX

<= ACK or NAK
```

The *text* portion consists of one or three ASCII characters which specify the current polled buffer or file and also the number of data records to be transmitted on a poll.

A Specifies the interactive buffer which holds only one record at a time (the most current record). The terminal will switch to buffered mode when the host is not polling the terminal.

C Specifies the interactive buffer which holds only one record at a time (the most current record). The terminal will go off-line and not allow any data to be entered when the host is not polling the terminal.

B9n Specifies the polled File 9 which will buffer multiple records (n = the number of records to be returned on a poll, maximum of 128 bytes).

When the terminal is switched to buffered mode, poll all the buffered transaction records out of the terminal before switching back to interactive mode. If the terminal is switched to interactive mode without polling all the buffered transactions from File 9 (transaction buffer), the terminal transmits one more buffered transaction when polled in interactive mode because of double buffering within the terminal microcode.

CAUTION:
When specifying either the interactive buffer or File 9 as the polled file, the following sequence is recommended. This prevents any conflict between the Command 8 and an input transaction which may be in progress.

The correct sequence is:

1. Take unit off-line
2. Select the current polled file
3. Put unit back on-line.

A transaction in progress when the unit is taken off-line is cancelled. The operator has to restart the transaction.

CMD 9X = Terminal Text or Data File Management

```
=> STX, addr, addr, 9X, text, RS, CS1, CS2, ETX
```

The 9X commands are used to initialize and manage up to 10 text or data files. Records in these files can be of variable length. Maximum length is 128 bytes with an ASCII record separator "RS" between records. This series of commands allows the host computer to perform the following:

Commands	Operation
9A	Request the status of files
9B	Initialize text files
9C	Load text into files
9D	Delete records from files
9E	Erase files
9F	Read text from files

The status should be queried (CMD 1) after each CMD 9X (text or data file command) to verify that the command was executed properly. An ACK response from the terminal, after receipt of any command means only that the terminal received the command properly. A status query (CMD 1) must be made to find out if the last command was executed properly within the terminal.

CMD 9A = Send File Status to the Host

```
=> STX, addr, addr, 9A, f, RS, CS1, CS2, ETX

<= STX, f, data, CS1, CS2, ETX

=> ACK or NAK
```

The f represents the file number (0—9).

The *data* portion consists of a fixed length string of ASCII characters which indicates the status of the specified file.

The format is haaahuuudt, where:

haaa Number of bytes reserved by previous command 9B (0001—P999)

h = 0—9 or A—P (10 through 25)
aaa = 000—999

For example:

haaa = B431 = 11431 bytes
haaa = E385 = 14385 bytes
haaa = N999 = 23999 bytes

huuu Number of bytes not used

h = 0—9 or A—P (10 through 25)
uuu = 000—999

d Text type specified

0 = compacted numerical text
1 = non-compacted alphanumeric text

t Type of text file

0 = non-polled file
1 = polled file

CMD 9B = Initialize the Specified Text File

=> STX, addr, addr, 9B, f, text, RS, CS1, CS2, ETX

<= ACK or NAK

The f represents the file number (0—9).

The *text* portion consists of a fixed length of ASCII characters which initializes a specified text file, reserves file space, and defines file characteristics.

The format is haaadt, where:

haaa File space to be reserved in bytes. Space for File 9 is defaulted to the unused portion of RAM by the terminal microcode (0000 would be valid entry for File 9).

h = 0—9 or A—P (10—25)
aaa = 000—999

d Type of text to be stored in the file

0 = Numeric text only; will be compacted (two numeric characters per byte including record separators)
1 = Alphanumeric text; will not be compacted

t Type of text file

0 = Non-polled text file (File 0—8)
1 = Polled data file (File 9 only)

One byte is reserved for an end-of-file marker in each file.

Non-polled text files are host-controlled files and may be loaded.

File 9 is a polled data file which may be loaded only under control of terminal microcode. Data in a polled data file cannot be erased by the host. This data remains in the file until removed by a poll from the host. Records in a polled data file can also be read by the host with a 9F command.

CAUTION:
Do not define File 9 as a numeric (compacted) file if alphabetic data is to be included in the transaction data record. If alphabetic data is stored in File 9 as a numeric file, certain combinations of alphabetic characters could be compacted as end-of-file markers. Terminal microcode would interpret this as the end of a file and prematurely terminate removal of data transaction records. This error condition is recoverable only by removing the power and communication connector (under the customer access panel). This destroys the contents of the RAM, and an initial terminal download must then be performed.

A text or data file cannot be reinitialized when data records exist in any polled data file. Attempts to do this will be rejected, and character number 8 of the status message will be set to a "6." Data records in all polled files must be removed by the host (polled) before reinitializing can be done. If it is necessary to reinitialize any text or data file, then all files have to be reinitialized.

CMD 9C = Load Text Files

```
=> STX, addr, addr, 9C, f, text, RS, CS1, CS2, ETX

<= ACK or NAK
```

The f represents the file number (0—8).

Command 9C is used to load text records into a specified non-polled text file. A maximum of 128 bytes may be transmitted at one time. Repeat the 9C command string to load more than 128 bytes to the file. Text records may be variable in length. An ASCII record separator RS must be entered between the records and at the end of the last record transmitted. The following format is an example:

9C31234RS123RS12345RS

This example sends text records 1234, 123, and 12345 to File 3.

Command 9C is valid only with non-polled files. Any attempt to load text to a polled file will be rejected, and status character number 8 will be set to "1."

CMD 9D = Delete Text Records From Files

```
=> STX, addr, addr, 9D, f Text, RS, CS1, CS2, ETX

<= ACK or NAK
```

The f represents the file number (2—7).

Command 9D is used to delete text records from a non-polled text file. More than one record can be deleted with one issue of command 9D. The limiting factor is a maximum of 128 bytes of record deletion at any one time. An ASCII record separator "RS" must be inserted between records and at the end of the last record transmitted. If the text record is not found in the file, then status character number 8 is set to "5." The following format is an example:

9D2JOHN(RS)MARY(RS)

This example deletes the records JOHN and MARY from File 2.

Command 9D is valid only for use with non-polled files. Any attempt to delete records from a polled file will be rejected, and status character number 8 will be set to "1."

When a record is deleted from a file, the file records are restructured to use the space left by the deleted record. Therefore, if record position is critical (such as prompt messages), this command should not be used.

CMD 9E = Erase Text Files

```
=> STX, addr, addr, 9E, f, RS, CS1, CS2, ETX

=> ACK or NAK
```

The f represents the file number (2–7).

Command 9E erases all text records in a specified non-polled text file. The following format is an example.

9E2

This example erases all text records in File 2.

Command 9E is valid only for non-polled files. Any attempt to erase a polled file will be rejected, and status character number 8 will be set to "1." Command 9E is not valid for use on a privileged file.

CMD 9F = Transmit Contents of Specified Text File to Host

=> STX, addr, addr, 9F, f, text, RS, CS1, CS2, ETX

- The f represents the file number (0–9).
- The text portion consists of the record number of the 128 byte block (000–999).
- The data portion consists of a block of 128 bytes that is transmitted to the host in response to this command.

An example format is:

9F3002

This example returns the third (128 byte) block from File 3. An ASCII carriage return character "CR" follows the last data record in the file.

If the specified text file has not been initialized, terminal response will be an ASCII cancel character "CAN" as follows. Status character number 8 will be set to "4."

<= STX, CAN, CS1, CS2, ETX

CMD A = Control Terminal (Ready) or (Not Ready)

=> STX, addr, addr, A, text, CS1, CS2, ETX

<= ACK or NAK

The *text* portion consists of an ASCII zero (0) to take unit out of service (NOT READY) or one (1) to put unit in service (READY). In the not-ready mode, the display will show a message OUT OF SERVICE - TIME. The terminal will effectively be out of service, and input transactions will not be accepted.

CMD B = Positive Response From Host

```
=> STX, addr, addr, B, text, RS, CS1, CS2, ETX

<= ACK or NAK
```

The *text* portion consists of a string of ASCII control and text characters used to control the audible tone and write messages to the display. STX and ETX characters cannot be in this text string.

The first text character controls the audible tone.

0 No tone, prompt timeout after timer B.

1 Sounds the tone defined for a valid (good) transaction; prompt timeout after timer B.

2 No tone, no prompt timeout.

3 Sounds the tone; no prompt timeout.

Choices 2 and 3 allow the prompt "CMD B" to remain on the display until another host command changes the display or the Clear key is entered from the keyboard.

Additional text characters define the display source:

#MSG(RS) Write message "MSG" to the display (message must end with record separator RS).

$frr Write record at "frr" to the display. Where:

f = File number (2–8)
rr = Record location (00–99) within the file

Examples:

B1#ACCEPT(RS)

This gives an audible tone for a valid transaction; it also writes "ACCEPT" to the display.

B1$403

This gives an audible tone for a valid transaction; it also writes the third record in File 4 to the display.

To prevent disturbing normal transactions, the use of CMD B is handled only during terminal idle cycles.

CMD C = Negative Response From Host

```
=> STX, addr, addr, C, text, RS, CS1, CS2, ETX

<= ACK or NAK
```

The *text* portion consists of a string of ASCII control and text characters used to control the audible tone and write messages to the display. STX and ETX characters cannot be in this text string.

The first text character controls the audible tone.

0 No tone, prompt timeout after timer B.

1 Sounds the tone defined for an invalid (bad) transaction; prompt timeout after timer B.

2 No tone, no prompt timeout.

3 Sounds the tone; no prompt timeout.

Choices 2 and 3 allow the prompt "CMD C" to remain on the display until another host command changes the display or the Clear key is entered from the keyboard.

Additional text characters define the display source:

#MSG(RS) Write message "MSG" to the display (message must end with record separator RS).

$frr Write record at "frr" to the display. Where:

f = File number (2—8)
rr = Record location (00—99) within the file

Examples:

C1#NOT ACCEPTED(RS)

This example gives the audible tone for an invalid transaction; it also writes "NOT ACCEPTED" to the display.

C1$403

This example gives the audible tone for an invalid transaction, it also writes the third record in File 4 to the display.

To prevent disturbing normal transactions, the use of CMD C is handled only during terminal idle cycles.

CMD D = Host Controlled Function Key

```
=>  STX, addr, addr, D, (func key parms), RS, CS1, CS2, ETX

<=  ACK or NAK
```

This command allows the host computer to initiate a data collection session with the terminal. Command D can have multiple function key parameters appended to it. These may include any of the following:

```
:0rr          Prompts Stored in File 8
:1llft        Fixed Length Badge
:2llft(appb)  Numeric Data
:3llft(appb)  Alphanumeric Data
:4llft        Alphanumeric Data or Badge
:6(text)      Prompt Sent from Host
;lld0         Variable Length Badge
```

Note: :6(text) is a new function key parameter for this command only. (text) is the prompt message to be displayed. Invalid (text) characters are (") , (:), (;) and (RS"), because they will be interpreted as parameter delimiters.

In interactive (8C) or interactive/buffered (8A) mode, command D will be accepted by the terminal (or commands B/C/2) to finish off a transaction. This provides the flexibility of using the terminal in a truly interactive mode, allowing the host to control each step in the data collection process.

Transactions created with CMD D contain a key id of "31" during buffered (8B9) mode or a key id of "81" during interactive (8C) and interactive/buffered (8A) modes.

```
=>  STX, addr, addr, D:010:11000:6Enter Dept:20500, RS, CS1, CS2, ETX

<=  ACK
```

This example generates a transaction from the terminal. A prompt is displayed (File 8, record 10) with a 10 character badge read expected. Next, "Enter Dept" is displayed with a 5 character numeric input expected. The resulting transaction may be polled from the terminal.

CMD E = Not Available

CMD F = Not Available

CMD G = Not Available

CMD H = Vary the Communications Turn-Around Time

```
=> STX, addr, addr, H, text, RS, CS1, CS2, ETX

<= ACK or NAK
```

The *text* portion consists of two ASCII numeric characters whose value can be from 01 through 99. This corresponds to milliseconds, and the new value will replace the previous or default value of 50 milliseconds.

Command H varies the terminal delay before transmit (turn-around time). This allows the user to optimize the system response time if the host turn-around time is less than 50 milliseconds or if the communications hardware has some timing variances.

Chapter 7. Keypad Operation, Advanced Features, and Control Protocol

Keypad Operation

Operational Chart

Key functions differ from the key captions depending on the mode of the terminal. The following table describes the key functions while in idle or data entry mode.

Key Captions	Idle Mode Function	Data Entry Mode Function
Numeric 0—9	No response	Puts numeric characters into a data entry field.
Special characters (See Note 2.)	No response	Puts special characters into a data entry field.
Enter	No response	Completes the transaction step in progress and advances to next step of the transaction; however, if this is the last step in the transaction, the Enter key operates as a transmit key.
Clear (See Note 2.)	No response.	If no characters have been entered into an editing field and the Clear key is activated, the terminal returns to idle mode. If one or more characters have been entered into an editing field and the Clear key is activated, the characters are erased and editing is resumed.
Function 1—26 or Alpha A—Z	Responds to the key definition for the corresponding key in File 0.	Puts alphabetic characters A—Z into the data entry field defined or initiates a transaction.
Function 27 or Fill Key	Responds to key definition (27) in File 0.	Blank fills an alphabetic field.
Function 28 or Space Key	Responds to key definition (28) in File 0.	Removes underscores and inserts blanks into an alphabetic key entry field.

Notes:

1. Initial power-up communications parameters are entered by using the numeric keypad as described in "Set-up Parameters and Download Sequence" on page 4-2.
2. Use of the Clear or a special character key involves a *shift* function. You must first press and release the Shift key; then press the Clear or character key.

   The shift function is reset after each operation. If another shifted function is needed, press and release the Shift key again before pressing the desired character. Shift is only valid for the numeric keypad.

Advanced Features

Writing Prompts or Messages Using Cursor Control

Cursor control or positioning of the data on the display or in the display buffer is accomplished by an escape sequence embedded in the data. An ASCII escape character (hex 1B) followed by two ASCII numeric characters (01—40) positions the leftmost character of the message to be displayed at display positions 01—40.

The escape sequence may be used multiple times within a message or prompt to selectively write and skip over selected fields on the display or in the display buffer. Cursor control can be used with any message or prompt from the host command 2, B, and C or the prompt message File 8.

Example of a prompt text field with two prompts:

(DC2)(1B)04PROMPT 1(1B)19PROMPT 2(RS)

Contents of the 40-character display and display buffer with the first prompt beginning in position 4 and the second beginning in position 19:

```
1    4                   19                  40
[    PROMPT 1            PROMPT 2            ]
```

Writing Prompts or Messages to the Display Buffer

Data written to the display times out after the interval in timer B (File 0 download parameter) has expired. An ASCII DC2 (device control 2) character used in the first character position of the text to be displayed causes the text to be written to the display buffer. Text written to the display buffer does not time out. Text remains on the display until the display buffer is cleared by pressing the Clear key or rewriting the display buffer.

The ASCII DC2 character can be used only in the first text character position of any message being displayed. The message could come from the host through commands 2, B, and C, or it could come from the prompt message file (8) by an operator pressing a function key.

Using Autofill with Key Entry Fields

The advanced feature optional parameters available on function key definitions :2 and :3 has an autofill option. If you use the advanced features, you must specify all four parameter characters.

- In a numeric key entry field (:2), if parameter "a" has been defined as a "1," pressing the Enter key changes all unused positions of the field to zeroes.
- In an alphanumeric (:3) key entry field, if parameter "a" has been defined as a "1," pressing the Enter key changes all unused positions of the field to ASCII spaces.

Using Cursor Controls with Key Entry Fields

When you use the advanced feature optional parameters (pp) in a (:2) or (:3) function key definition, the left-most position of the field will be at (pp) 01—40 display position.

Writing Key Entry Fields to the Display Buffer

The last advanced feature parameter (b) is used to write a key entry field to the display buffer. The primary use of the display buffer is to store data until another function key or badge initiated transaction sends the data to the host computer.

- When b = 0, the keyed data is displayed but not entered in the display buffer. Data written to the display times out after the interval in timer B (File 0 download parameter) has expired.
- When b = 1, the keyed data is displayed and entered to the display buffer.

If a do-not-transmit (;6) function key definition has preceded this (:2) or (:3) function key definition and you do not write the key entry field to the display buffer, the data is lost as soon as the operator presses the Enter key to complete the field.

CAUTION:
The user is responsible for implementing controls to ensure data integrity when using these advanced functions. You can overwrite data, lose data, and the transaction you create may not be what you expected.

Communications Control Protocol 1283, Version 1.0

This section describes the IBM-designed 1283, Version 1.0, Communication Line Control Protocol which is the basis for communication between the host computer and attached IBM 7525 Data Collection Terminals. The protocol uses asynchronous serial ASCII characters for communication between the host computer and various terminals including the 7525 Terminal.

Reserved ASCII Characters

ASCII characters reserved for use in the text portion of commands are listed in the following table. They are not allowed in encoded badge data.

ASCII	Hex	Description
ACK	06	Positive acknowledgement
ETX	03	End of text
NAK	15	Negative acknowledgement
STX	02	Start of text
RS	1E	Record separator
CAN	18	Cancel
DC1	11	Device control 1
DC2	12	Device control 2
ESC	1B	Escape sequence

The "ASCII Code Chart" on page 7-5 gives a complete listing of ASCII codes.

ASCII Code Chart

HEX High Order

8	9	A	B	C	D	E	F	Parity = 1
0	1	2	3	4	5	6	7	Parity = 0

HEX Low Order

0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

ASCII Codes

7525 Data Collection Terminal Response Time

When a 7525 Data Collection Terminal receives a command or a poll, it responds within a time referred to as "turn-around time."

- Turn-around time varies in relation to the processing time needed for handling data and transactions within the terminal.
- The default minimum turn-around time is 50 milliseconds.
- The maximum turn-around time is 100 milliseconds (if response is possible).

Command H is sent to the terminal to change the minimum turn-around time. Performance may be optimized to the communication hardware and/or the host system requirements.

Appendix A. Terminal Detected Errors

Terminal Error Messages

Terminal microcode will display error messages when error conditions are detected as shown below. These error messages are not stored or sent to the host computer. The messages will be displayed for a period of time as defined by timer B download parameter in File 0, record 00.

E01 There was an invalid attempt to use a file that has not been initialized, such as attempting to load an uninitialized file.

E02 There was an invalid attempt to use a prompt record which does not exist in File 8.

E03 A function key parameter table format error was detected.

E04 Invalid prompt format:

- Optional cursor positioning value greater than display size (40 characters), *or*
- More than one DC2 in the prompt string, *or*
- Optional cursor positioning value set up as a non-numeric value.

E05 Transaction created contains more than 128 characters.

File 8 Error Messages

The error message text of File 8 records is customer defined. The following list will assist in identifying and correcting the error condition.

File 8 record	Error message or condition	Host error record created
01	Timeout prompt	
	Displayed if the operator does not complete a transaction step or press a key during data input before the allotted time in timer C download parameter, File 0, record 00, bytes 19 and 20 has expired. See Table 5-4 on page 5-8.	No, but the aborted transaction will have to be restarted
02	Invalid length prompt	
	An attempt to read a badge or a bar code of a different length than was specified in a function key definition or the fast clocking badge length parameter in File 0, record 00, byte 14 and 15. See Table 5-3 on page 5-7 and "Function Key Parameter Descriptions" on page 5-9.	Yes, 4XX record created in File 9 only if File 0, record 00, byte 7 is a 1 or 2
03	Invalid data prompt	
	Displayed if an attempt to validate data entered against a text file failed.	Yes, 4XX record created in File 9
	A magnetic badge read failure will also cause this prompt to be displayed if File 0, record 00, byte 7 is set to a 2. See Table 5-3 on page 5-7.	Yes, 9XX record created in File 9
04	Invalid operation prompt	
	An operator reading a badge or entering data in a different sequence than was specified in the function key definition. See "Function Key Parameter Descriptions" on page 5-9.	No

Appendix B. Configuration Guides and Forms

Features, Options, and Accessories

Make photocopies of pages in this appendix to specify features, quantities, and custom keypad overlays when planning your configuration.

Table B-1. 7525 Data Collection Terminal features order form

Models and Standard Features	**Quantity**
Model 001 • 40-character, alphanumeric, LCD display • Keypad with numeric keys, nine special characters and 28 function/alphabetic keys • Host communication using RS422 protocol • Bar code wand connector • 25 Kbytes of user random access memory (RAM) • Time-of-day (TOD) clock chip • Capacitive data-loss backup for TOD and RAM	
Model 002 Includes all the features of the Model 001 plus a visible-light bar code slot reader.	
Model 003 Includes all the features of the Model 001 plus an infrared light bar code slot reader.	
Model 004 Includes all the features of the Model 001 plus a magnetic slot reader.	
Optional Features	**Quantity**
15F8904 - Wall mount (100–125 Vac)	
15F8905 - Desk mount (100–125 Vac)	
15F8906 - Wall mount (200–240 Vac)	
15F8907 - Desk mount (200–220 Vac)	
15F8780 - Bar code wand (low-resolution, visible-light)	
15F8778 - Bar code wand (high-resolution, visible-light)	
15F8784 - Bar code wand (low-resolution, infrared)	
15F8782 - Bar code wand (high-resolution, infrared)	
15F8786 - RS422 communication cable	

Keypad Overlay Printing and Colors

It is your responsibility to design and obtain any custom overlays required by the application or operator use of the 7525 Data Collection Terminals. The following forms are provided to help in defining the custom overlay. Use photocopies of form sheets in this appendix to specify keypad key printing, colors, and styles.

Use "Layouts for Keypad Overlay" on page B-3 to show the keys you want to use for your application. Cross out any keys that are not used. Write in numbers, alphabetic characters, or function key legends you require for your application. Also refer to the sheet for allowable key assignments. You may specify special function key symbols or printing as desired.

Numeric keys (the block of keys on the left side of the keyboard) may be configured as calculator or telephone key layout. The telephone key layout is shown on the layout sheet form. Any combination of alphabetic keys and function keys may be specified.

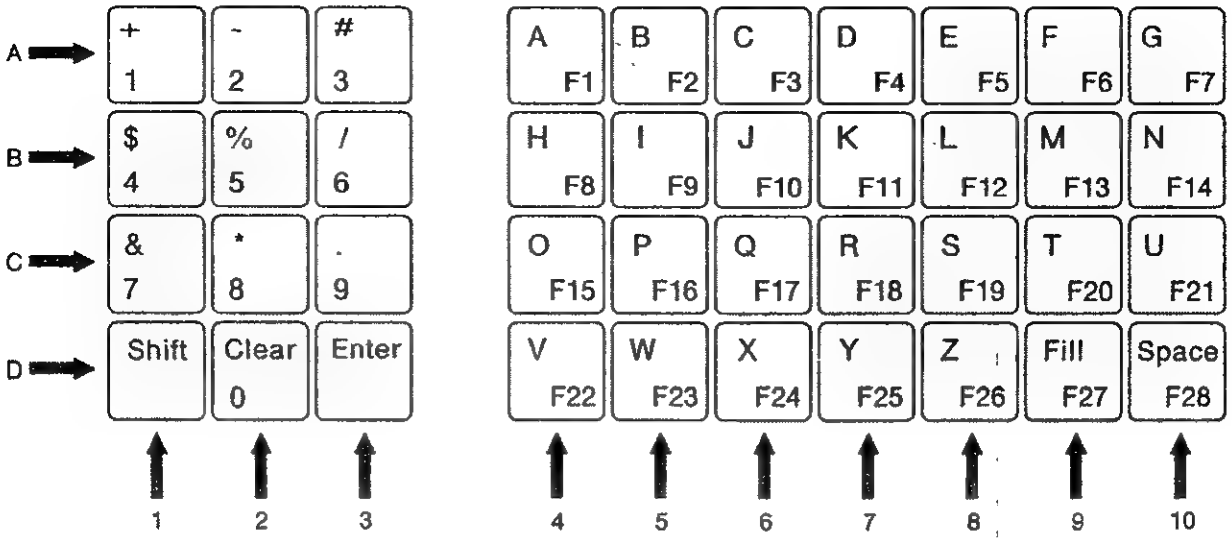
Pantone[1] colors, color numbers, and key styles may be specified for key tops on "Keypad Style/Color Order Forms" on page B-5. Keys may be arranged in blocks of different colors.

Use the form "Special Overlay Instructions" on page B-7 to specify background color and any special overlay instructions. Provide camera-ready copy for any special logos or symbols shown in key squares or at any other location on the key assignment chart.

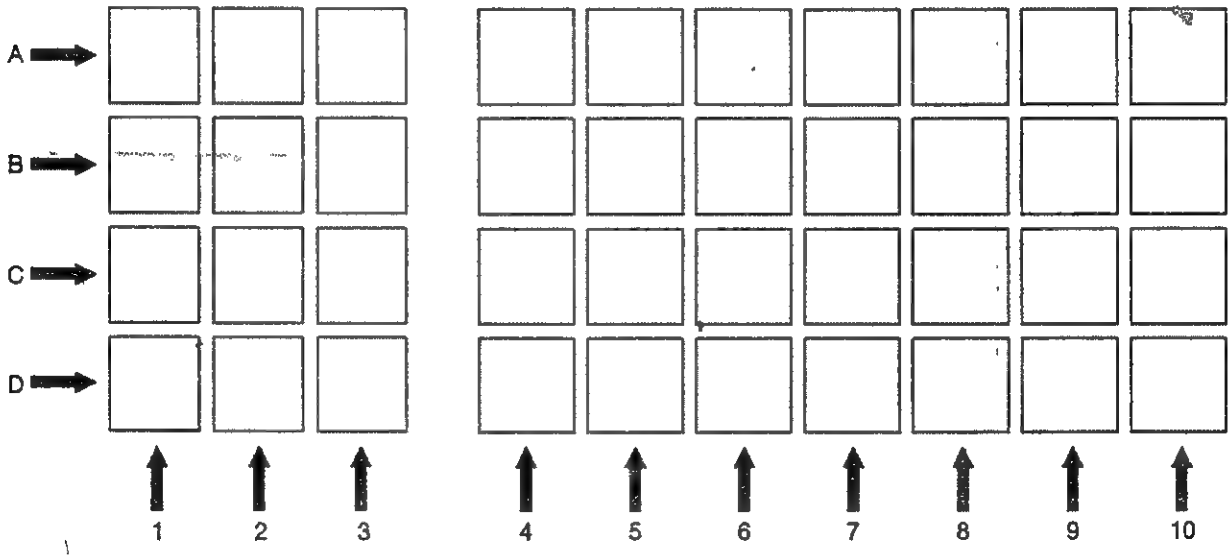
Dimensions for a custom keyboard overlay are shown in "Dimensions for Keypad Overlay" on page B-4.

[1] Pantone, Inc.'s check-standard trademark for color reproduction and color reproduction materials.

Layouts for Keypad Overlay



Key captions and styles

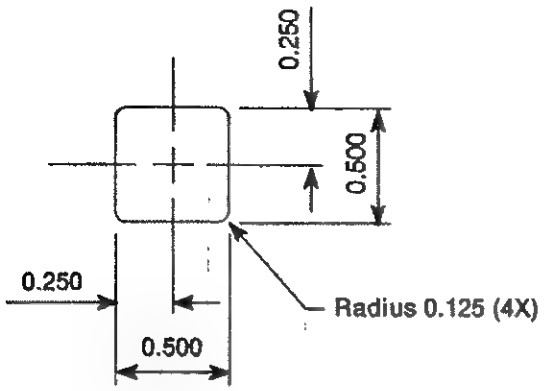


1. Show keytop print wanted in keysquares.

2. Show position of special markings in keysquares such as logo(s) or special symbol(s).

Key assignments

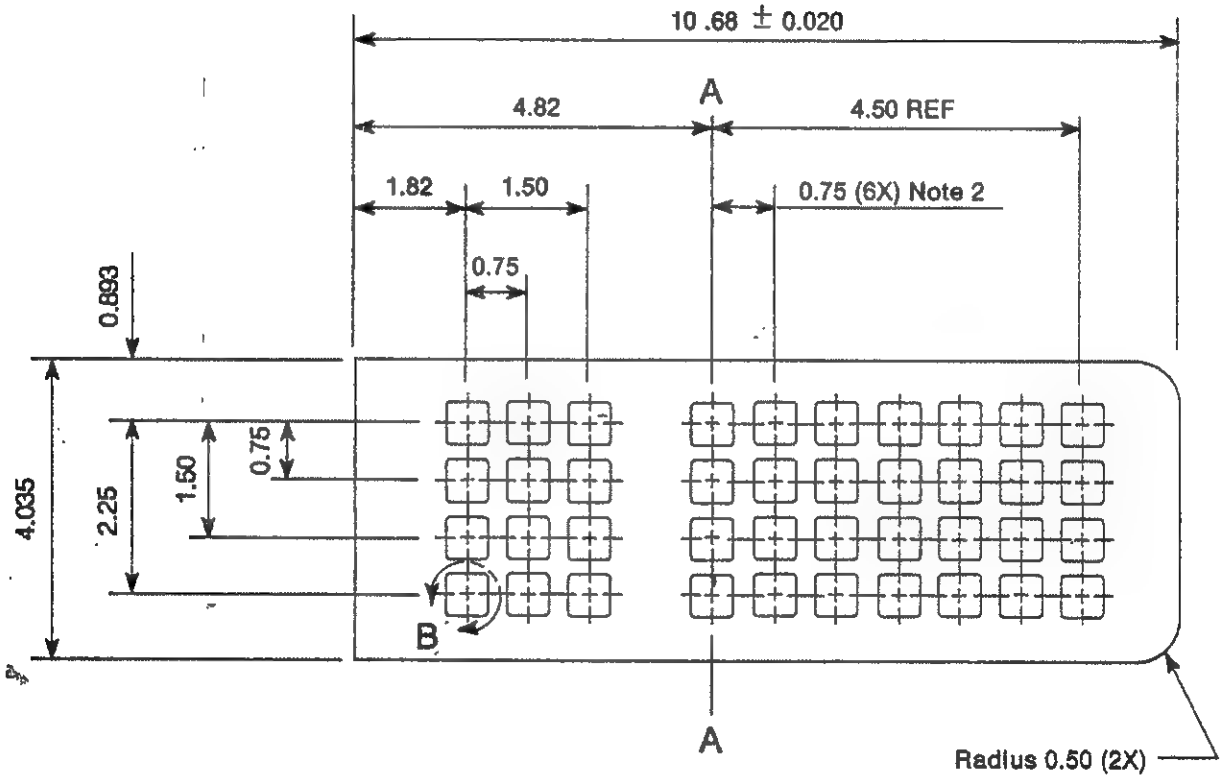
Dimensions for Keypad Overlay



Detail B

Notes:

1. Material to be 0.010 ± .002 thick polycarbonate
2. (6X) 0.75 May vary ± 0.01 from A-A



Keypad Style/Color Order Forms

Numeric Key Order Form

Key	Key Style	Key Background		Key Printing		Key Border	
Coordinates	A or B	Color	Number	Color	Number	Color	Number
A1							
A2							
A3							
B1							
B2							
B3							
C1							
C2							
C3							
D1							
D2							
D3							

Alpha/Function Key Order Form

Key	Key Style	Key Background		Key Printing		Key Border	
Coordinates	A or B	Color	Number	Color	Number	Color	Number
A4							
A5							
A6							
A7							
A8							
A9							
A10							
B4							
B5							
B6							
B7							
B8							
B9							
B10							
C4							
C5							
C6							
C7							
C8							
C9							
C10							
D4							
D5							
D6							
D7							
D8							
D9							
D10							

Special Overlay Instructions

Color of background area

Indicate Pantone color and number below for background. This area does not include keytop color, style, or symbols.

Special instructions

Indicate any special overlay requirements below. Provide camera-ready copy for any special symbols or logos positioned on the background area.

Appendix C. Machine Readable Media Standards

This section contains standards and specifications for machine readable media applicable to the IBM 7525 Data Collection Terminal.

Note: Slot readers are factory adjusted for reading badges which have a thickness of 0.028 inch $\pm$ 0.0028 inch.

Bar Code Reader and Badge Guidelines

Bar Code Reader Selection

There are a number of choices in selecting a bar code reader. A bar code slot reader or wand may be selected with either a visible-red (660 nanometer range) or infrared (880 nanometer range) illumination source. With a bar code wand there is the additional choice of a high-resolution aperture (6 mm) or a low-resolution aperture (10 mm).

Bar Code Reader

- If the bar code labels contain carbon-based inks, either a reader with infrared or visible-red light illumination may be used.
- Bar code labels using dye-based inks or thermal transfer processes require a bar code reader with visible-red light illumination.
- When using a bar code label with black print and a black overlay (background), an infrared reader is required.

Bar Code Wand

- For applications using high-density bar code labels, a bar code wand with a 6 mm aperture is preferred.
- A bar code wand with a 10 mm aperture is intended for applications requiring low-density bar code labels such as those printed with a dot-matrix printer.

Bar Code Badge Label Guidelines

For the bar code slot reader and wand to operate properly, follow these recommendations:

- The bar code symbol should be dimensionally correct with adequate quiet zones. Bar and space tolerance specifications differ from one symbology to the next. The quiet zones are areas clear of stray marks which immediately precede the first bar and immediately follow the last bar. Each of these quiet zones should be at least 0.25 inch wide or 10 times the narrow element width, whichever is greater.
- The white or space/quiet zone reflectance should exceed 50% at the illuminating scanning wavelength.
- A high print contrast ratio allows for sharper black-to-white transitions and signal amplitude. A print contrast ratio greater than 0.75 is recommended for optimum read rates.
- When creating a badge with labels and pictures, the scanning surface should be kept flat. This may require the use of cutouts or inserts in the badge material.

- When using laminated badges, make sure that the plastic overlay does not violate dimensional and reflectance requirements.
- Any black or red overlay (background) which prevents the bar code label from being seen or photocopied should be transparent to the illumination wavelength of the selected reader. To ensure readability, the above requirements on print contrast ratio and reflectance must be adhered to.

Bar code

The 7525 Terminal can read Code 3 of 9, Interleaved 2 of 5, and UPC/A bar codes. No attempt is made to fully describe all the codes in this manual. The most popular bar code for industrial users is Code 3 of 9.

Code 3 of 9 Bar Code Description

The Code 3 of 9 bar code is a variable length, discrete, self-checking, bidirectional, alphanumeric bar code. Its character set contains 43 characters: 0−9, A−Z, −, ., $, /, +, %, and space. Each character is composed of nine elements: five bars and four spaces. Three of the nine elements are wide (binary value 1) and six elements are narrow (binary value 0). Spaces between characters are not significant; therefore, Code 3 of 9 bar code is discrete. A common character (*) is used for both start and stop delimiters. An example of a Code 3 of 9 bar code message containing the string "ABC," is shown in Figure C-1 on page C-4 .

Code Configuration

A message consists of any number of data character symbols enclosed between two start/stop code characters.

Reflectivity

Reflectivity measurements are to be made at a wavelength of 900 nanometer (nm) with a bandwidth to the 50% level of 40 nm or less. Incident irradiation should be 45° to the normal and reflected flux collected with a 15° angle centered on the normal. Reflectance values are referred to a magnesium oxide or barium sulfate standard at 100%. The reflectance of the background and white spaces within the code should exceed 70%.

Contrast

The print contrast signal (PCS) is defined as:

$$PCS = \frac{Rw - Rb}{Rw}$$

where Rw is the reflectance from the white spaces and Rb is the reflectance from the dark bars. PSC should exceed 65%.

Code Density and Dimensions

The Code 3 of 9 bar code can be printed at various densities to accommodate a variety of printing and reading processes. The significant parameters are the nominal width of the narrow elements (unit size) and the nominal ratio of wide to narrow elements. The allowable range of minimum unit size and wide-to-narrow ratio is:

Minimum nominal unit size	0.0075 inch (0.19 mm)
Maximum nominal unit size	0.0400 inch (0.571 mm)
Nominal wide-to-narrow ratio	2.2:1 to 3.0:1 for codes whose unit size is less than 0.02 inch
	2.0:1 to 3.0:1 for codes whose unit size is more than 0.02 inch

The bar code height can vary to suit specific reading and printing requirements. For hand scanning, the minimum bar height is 0.25 inch (6.4 mm) or 15 percent of the bar code length, whichever is greater. For noncontact reading, the minimum bar height is 1.25 inches (31.8 mm) or 25 percent of the bar code length, whichever is greater.

Inter-Character Gap

The minimum gap between characters is the same as the dimension of a minimum unit of (x - t). The maximum inter-character gap width shall be no more than three times the width of a wide element, or 0.060 inch (1.5 mm), whichever is greater.

Margins (Quiet Zones)

The minimum left and right margins shall be 10 times the width of one narrow element, or 0.10 inch (2.5 mm) whichever is greater. For hand scanning, the margins shall be at least 0.25 inch (6.4 mm).

Spacing Between Bar Code and OCR-A

The minimum spacing between the bar code and the OCR-A shall be at least 0.10 inch (2.5 mm).

Spots and Voids

Spots and voids small enough to be contained entirely within a 0.003 inch diameter circle or which occupy no more than 25 percent of the area of a 0.006 inch circle are allowed and should not have an adverse effect on properly configured reading equipment. Larger spots or voids reduce the first-read-rate performance depending on the size of such larger spots and voids.

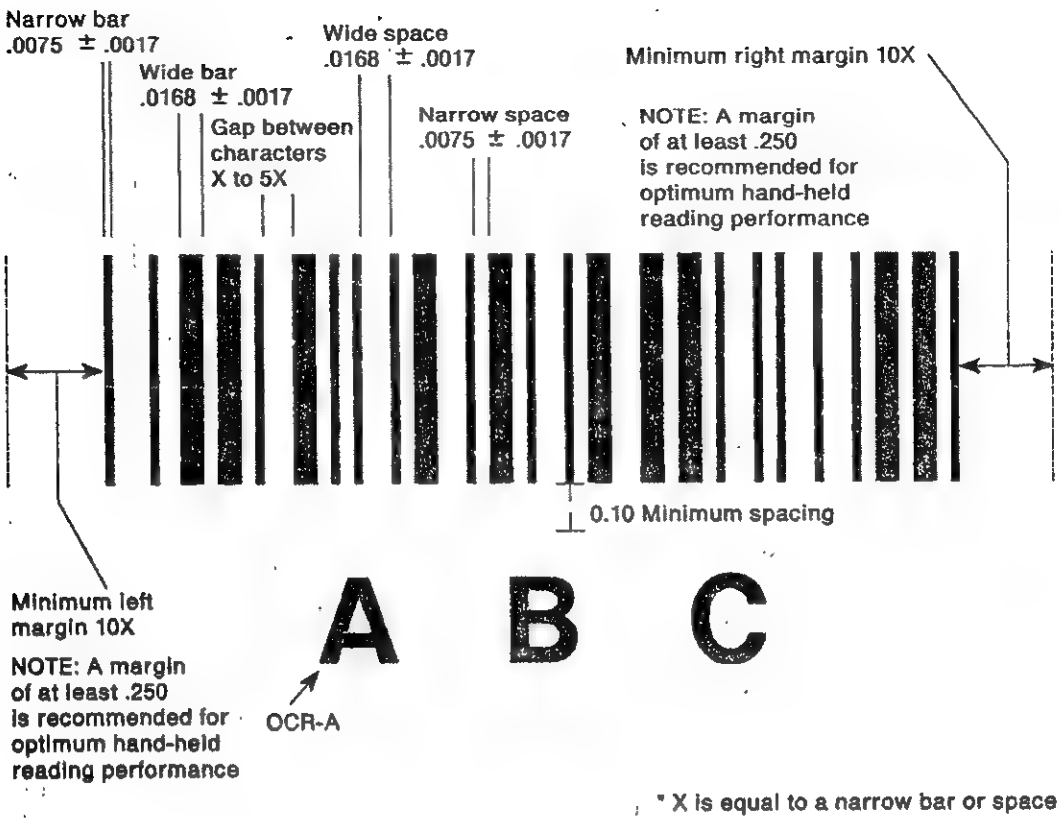


Figure C-1. Standard Code, 9.4 Characters Per Inch Density (Enlarged)

Bar Code Badge Specifications

Scope

This section describes the physical specifications for bar code badges that *have not* been embossed, including general characteristics, dimensions, printing specifications, and physical location of code.

Badge Material

Badges shall be made of laminated plastic or a material having equal or better physical and electrical performance characteristics. Bar code shall be embedded in the plastic not more than 0.005 inch, ± 0.001 inch from the reading surface of the badge, and protected by a clear layer of laminated plastic material.

The laminated plastic material on the back of the badge covering the bar code shall be capable of passing light in the range of 600—800 nanometers.

Badge Finish

The front of the badge shall have a gloss finish. The back of the badge shall have a matte finish.

Badge Dimensions

- The dimensions of the badge shall be as shown in Figure C-2 on page C-7.
- Badge thickness (not including edge burrs) shall be 0.028 inch, ± 0.0028 inch.
- The corners shall be rounded to a radius of 0.125 inch.
- Edge burrs normal to the face of the card shall not exceed 0.033 inch.

General Characteristics

Flammability

The badge shall be flame resistant and self-extinguishing in still air.

Toxicity

The badge shall present no toxic hazards during normal use.

Resistance To Chemicals

The badge shall be resistant to chemical effects arising in normal handling and use.

Temperature Stability

The badge shall remain structurally reliable and usable at environmental temperatures between −35° C and +50° C (−30° F and + 122° F).

Note: Environmental temperatures does not mean card temperature, but the environment in which the card is used.

Humidity

The badge shall be reliably usable at a relative air humidity between 5% and 95%, with a maximum wet bulb temperature of 25° C (77 F).

Ultraviolet Light

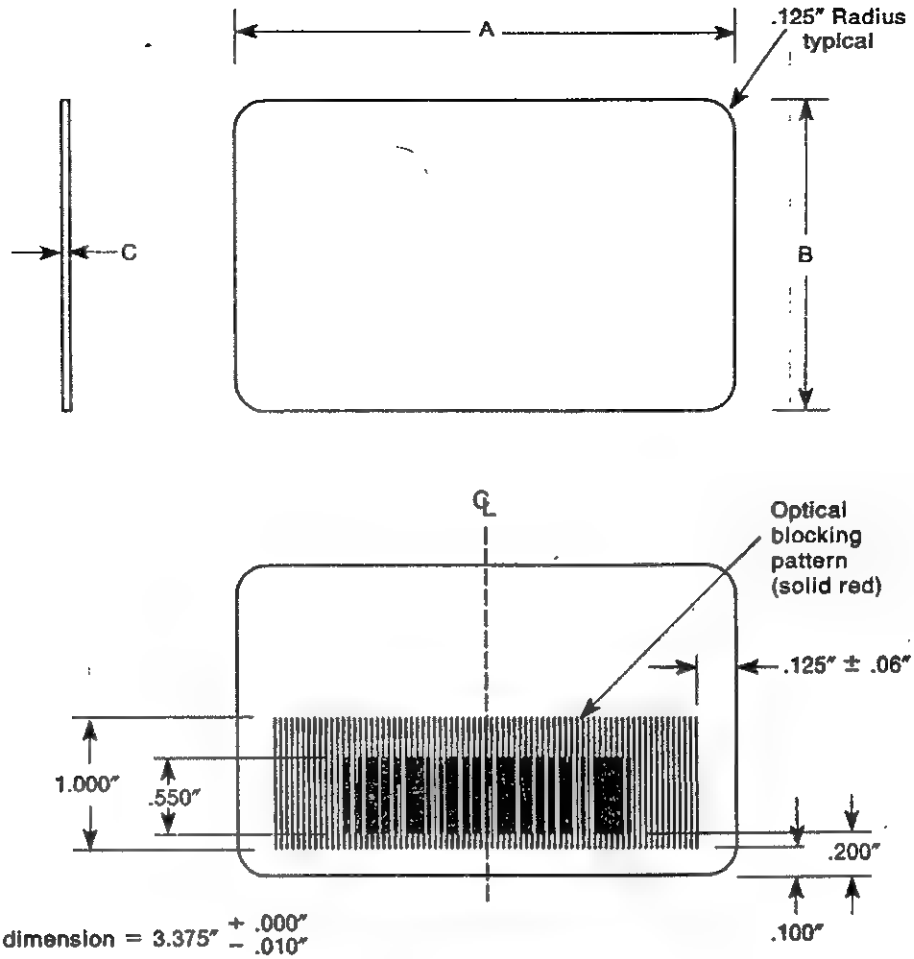
The badge and its printed text shall resist ultraviolet light encountered under conditions of normal use.

Printing Specifications for Bar Codes

- Printing on the back of the card shall conform to specifications shown in Figure C-2 on page C-7.
- Bar codes printed on white core stock with PMS Process Black-2C may use an optional blocking pattern with red printing which is optically transparent to light in the 600—800 nanometer wavelength range. This blocking pattern is prevents duplication of the badge on a standard (black and white) copier at standard settings.
- The print contract ratio (PCR) for bar code shall be greater than 0.75 for light in the 600-800 nanometer wave length range:

$$PCR = \frac{R(plastic) - R(ink)}{R(plastic)} > .75$$

Where R = Reflectance



"A" dimension = 3.375" $^{+\ .000"}_{-\ .010"}$

"B" dimension = 2.125" ± .002"

"C" dimension = 0.028" ± .0028"

Blocking pattern height = 1.000"

Bar code printing to be centered on the badge as shown with respect to left and right margins.

Figure C-2. Badge Printing Specifications

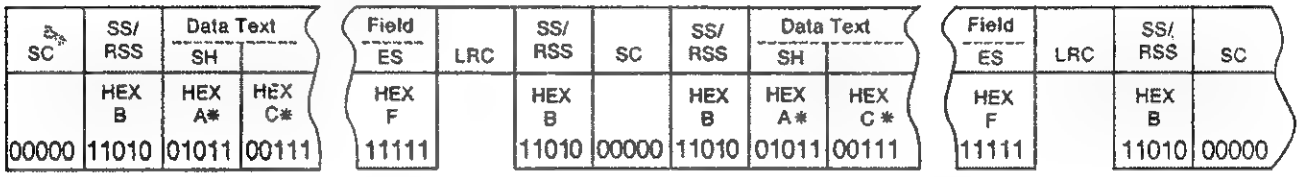
Magnetic Stripe Code

Description

The magnetic stripe code employed by the 7525 Data Collection Terminal is the ABA Track 2 or full-width, NRZ (non-return to zero) F2F encoding scheme. The magnetic slot reader is bi-directional (reads in either direction).

Format

The format of the magnetic stripe is:



* See "Data Text" on page C-9.

Control Characters

The following 4-bit hexadecimal control characters are used:

- Start Sentinel/Reverse Start Sentinel (SS/RSS), represented by hex B, is used at the beginning of each data text section and immediately following the LRC character.
- Secure Header (SH) (Optional). The secure header, represented by hex A, is the first character of a *secured* data text. It is not present in an unsecured data text. When the SH character is present, the associated data text section should not be printed, displayed, or edited by the entry device.
- Character Set Header The character set header is the first or second character of text (depending on the secure header). It identifies the character set in use.

  See "Data Text" on page C-9 for details of secure header and character set header use.
- End Sentinel (ES), represented by hex F, is used at the end of each data text section. The ES character is always followed by the LRC character.
- Synchronization Character (SC), represented by all zeros without parity, allows the hardware to synchronize with the characters to be read. There is a minimum of four synchronization characters at the beginning and end of the encoded message and a minimum of three between redundantly encoded messages.
- Longitudinal Redundancy Check (LRC) character is calculated from SS/RSS and through ES. SS/RSS following LRC is not included in the LRC calculation. When the magnetic stripe is scanned in the reverse direction, SS/RSS is read after SC and before LRC.

  The four bits of the LRC (weighted 1, 2, 4, and 8) are such that there is even parity on all the corresponding bits beginning with SS/RSS through ES. The parity bit of the LRC character is the odd parity of the four LRC data bits. It is not the even parity of the parity bits of the data characters.

Data Text

Message length can range from one to a maximum of 99 hexadecimal characters.

The 7525 uses two character sets, numeric and 63-character. The first two characters of each data text field define the character set and secure status as follows:

##	Numeric character set, not secure
A#	Numeric character set, secure
CA	63-character set, not secure
AC	63-character set, secure

Where:

is hex value 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, or D (field separator).
A is the hex value A and, if first, represents the secure header.
C is the hex value C and is used to identify 63-character set records.

Character Sets

Numeric Character Set

The numeric character set uses 10 hexadecimal numbers plus hex D for coding information. In addition, hex A is used for the secure header, while hex B and F are used as control characters. Hex E and C are *not* used.

The space character is recorded on the stripe as a hex D and is sometimes referred to as a field separator character.

The 7525 terminal converts the codes on the stripe to the ASCII codes as shown in the following numeric character set table.

Magnetic recording	0	1	2	3	4	5	6	7	8	9	D
ASCII	30h	31h	32h	33h	34h	35h	36h	37h	38h	39h	20h
Printable graphic	0	1	2	3	4	5	6	7	8	9	Space

63-Character Set

The 63-character set consists of numerics, alphabetics, and special symbols. Recording of each character requires two 4-bit plus parity hexadecimal characters. Numerics are packed in pairs, either an even number of numerics in any continuous string, or an odd number of numerics with the filler hex character A.

The hex alphabetical characters A, C, D, E, are used with the hex numbers to code information. Hex B and F are also used but only as control characters.

The 7525 terminal converts the codes on the stripe to the ASCII codes as shown in the following 63-character set table.

Read the *column* first and then the row; for example:

- Hex C1 on the stripe represents the letter A.
- Hex 38 represents the two decimal numbers 3 and 8 (paired or packed).
- Hex 3A represents the single decimal 3.

  In the 63-character set, non-numeric characters must start in an odd number location. Therefore, odd numeric fields must be padded with a trailing hex A in order to preserve this alignment of data on the stripe.

Hex	Column															
Row	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	10	20	30	40	50	60	70	80	90					\	
1	01	11	21	31	41	51	61	71	81	91			A	J	/	
2	01	12	22	32	42	52	62	72	82	92			B	K	S	
3	03	13	23	33	43	53	63	73	83	93			C	L	T	
4	04	14	24	34	44	54	64	74	84	94			D	M	U	
5	05	15	25	35	45	55	65	75	85	95			E	N	V	
6	06	16	26	36	46	56	66	76	86	96			F	O	W	
7	07	17	27	37	47	57	67	77	87	97			G	P	X	
8	08	18	28	38	48	58	68	78	88	98			H	Q	Y	
9	09	19	29	39	49	59	69	79	89	99			I	R	Z	
A	0A	1A	2A	3A	4A	5A	6A	7A	8A	9A			SP	&	-	
B																
C	¢	!		:	<	*	%	@								
D	.	?	,	#	(	)	_	'								
E	|	⌐	?	"	+	.;	>	=							Test	
F																

Notes:

1. Hex patterns with B or F in the first or second position are invalid. Hex B and F are used only as control characters.
2. Hex EE is a test stripe code.

The following example shows how a printable graphic would appear in ASCII in the 7525 and on the magnetic stripe.

Note: In the magnetic recording, the leading hex CA in the example is the 63-character set, non-secure header.

Magnetic recording	CA	C1	1	2	3A*	C2	3C	4	5	C3
ASCII		41h	31h	32h	33h	42h	3Ah	34h	35h	43h
Printable graphic		A	1	2	3	B	:	4	5	C

* Hex bit pattern 3A represents decimal 3. The trailing A pads the field to ensure an odd number starting location for the non-numeric data that follows (in this case an alpha character "B").

Error Detection

An odd parity bit for all encoded characters is used and parity is assigned to each 4-bit character.

Additionally, an even-bit-parity longitudinal redundancy check (LRC) code is at the end of the data text section.

Redundant Encoding

Data text fields may be repeated as many times as they will fit on the stripe. This is called redundant encoding.

The magnetic reader logic in the terminals accepts the first data field for which:

- Character parity is set, *and*
- The LRC character is correct, *and*
- There is at least one hexadecimal character between SS and ES.

Encoding Density

Encoding density on the stripe is 75 or 128 bits per inch (2.6 or 5.0 bits per millimeter.)

Encoding Specifications for Magnetic Badges

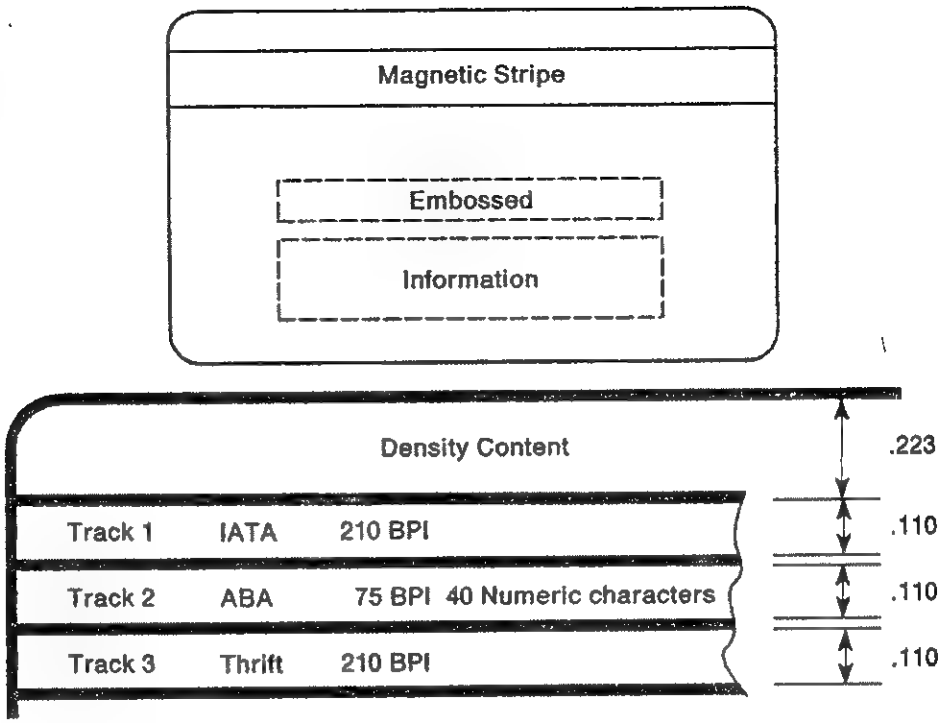


Figure C-3. Magnetic Badge Encoding

Magnetic Media Size Requirements

The maximum width of the magnetic slot is 0.036 inch. The badge or card thickness must be less than the maximum width of the slot. The size of the badge may vary if the distance from the magnetic strip encoding to the top edge of the badge or card is kept at the ABA Track 2 standard (0.353 to 0.463 inch from the top of the card).

Surface Distortions

The magnetic read head must be in contact with the magnetic stripe while reading. Any surface distortions, irregularities, or raised areas must not interfere with the reading of the magnetic stripe. Any embossing, pictures, or other magnetic material must be kept at least 0.25 inch from either the top or bottom edge of the magnetic stripe.

Magnetic Media Quality

The quality or durability of the magnetic stripe media can vary considerably. A time and attendance badge that will be used several times a day for years will have a high-durability requirement. A shop-floor document that is assigned to a job or product may not have a high-durability requirement. The requirements of the application determine the quality of the magnetic media. The customer is responsible for ensuring that the quality of the magnetic media is maintained at the level required for the application.

Appendix D. Physical Installation and Typical Configurations

Physical Planning

IBM does not provide for installation of 7525 Terminals. Physical planning is vital to bring about a smooth and successful installation of your 7525 Data Collection System. This means selecting and laying out the physical location for each component of your system, specifying and providing for all cabling, connector, and electrical requirements. Devising a realistic and orderly schedule for accomplishing each task leads to a successfully completed installation.

In the early stages of planning, your company should designate a project leader to be responsible for all phases of the 7525 installation. Your company should also designate parties to be responsible for carrying out specific tasks in the installation schedule. You may chose your own personnel or outside contractors.

In planning the installation, your company should clearly understand its responsibilities, which include:

- Perform the required physical planning. Select and layout the location for each desired component. Assure that you have not exceeded the 4000-foot distance limitation on individual RS422 cable runs.
- Order and stock spare 7525 terminals. The exact number of spare terminals depends on the critical nature of your application and the total number of terminals in your system.
- Ensure proper functioning of the power receptacle and power at the screw terminal connections in the wall mount assembly.
- Install all cabling and connecting hardware between the terminals and the host computer.
- Ensure safety and conformance to all your local and municipal electrical and building codes.
- Provide and install mounting hardware for each terminal.
- When the terminals arrive, it is your responsibility to unpack and mount each data collection terminal.

Grounding

A common cause of terminal problems is improper grounding. The 7525 terminal power cord or wall mount power ground must be connected by a separate grounding wire to a grounding terminal bar in the branch circuit distribution panel. The grounding terminal bar must be connected to the service entrance grounding electrode.

The grounding wire must be an insulated, non-current-carrying conductor of at least the same size as the branch circuit feeder. While the grounding wire can be run in the same conduit as the other wires, it must be kept electrically isolated from the neutral wire (the center tap of the service transformer). The grounding wire and this neutral wire are common only at the service entrance grounding electrode.

If it is not practical to run the grounding wire back to the service entrance (as in a tall building), you can use an exposed section of a cold water main that has continuous metal to ground. Building steel or grounding rods may be used if no other grounding means are available. All grounds must be tested. Check applicable national and local safety standards.

CAUTION:
This product is equipped with a 3-wire power cord and plug for the user's safety. Use this power cord in conjunction with a properly grounded electrical outlet to avoid electrical shock.

Lightning Strike Protection

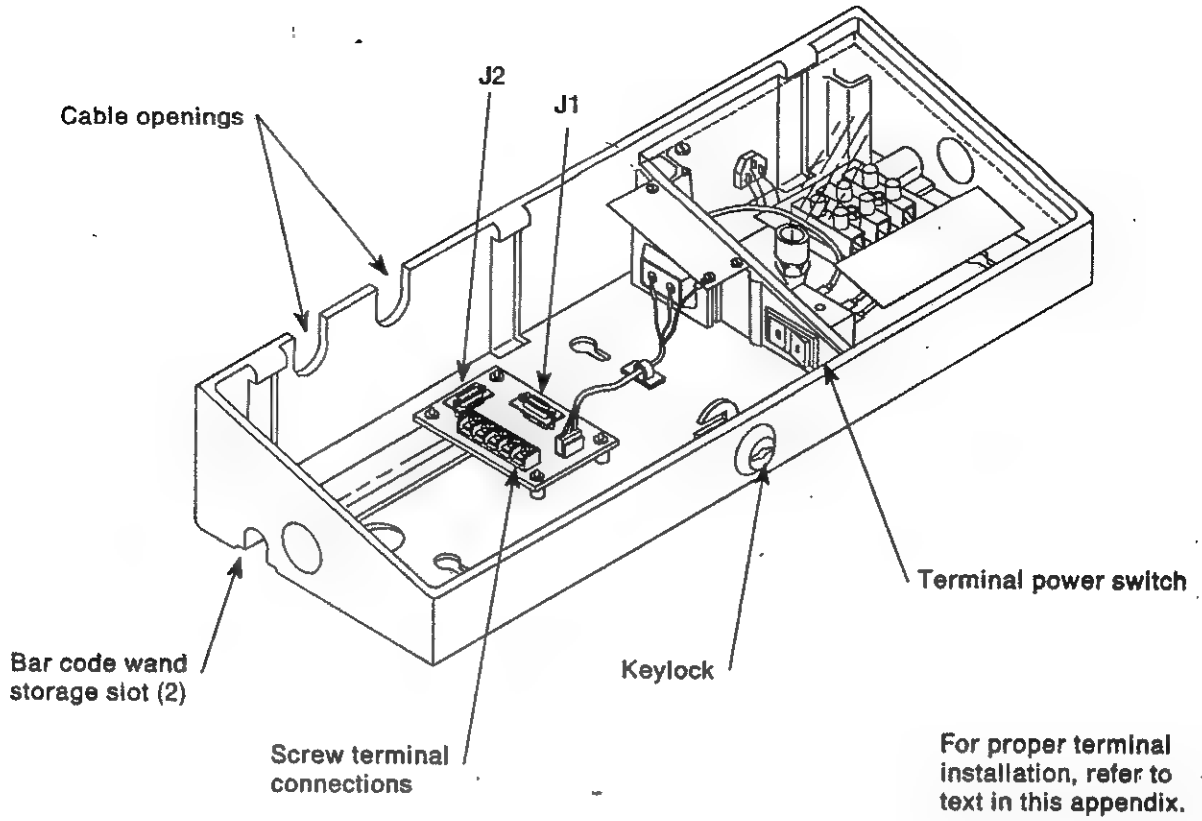
Signal cables that enter or exit a building must have appropriate lightning strike protection installed at the point the cables enter or exit a building.

Terminal Mounting and Dimensions

The 7525 Data Collection Terminal can be ordered with desk mounting (Figure D-1 on page D-3) or wall mounting (Figure D-2 on page D-4). Connection of cables to desk or wall mounts is shown in "Typical RS422 Terminal Wiring Schematics" on page D-9.

A Berg (4x2) connector plug must be attached to each RS422 line that attaches to a port on an IBM Series/1 MultiFunction Attachment Feature Card (1310). Refer to "Multifunction Attachment Card Jumpers" on page 3-4, and "Multifunction Attachment Card Ports" on page 3-5.

Refer to "Typical RS422 Host Communication Wiring Schematic" on page D-8 to wire a 9-pin, D-Shell connector for attachment to the RS422 Communications Adapter Card in an IBM Personal Computer.



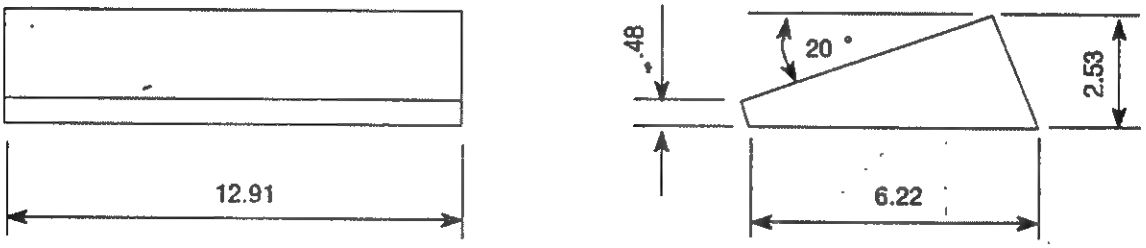
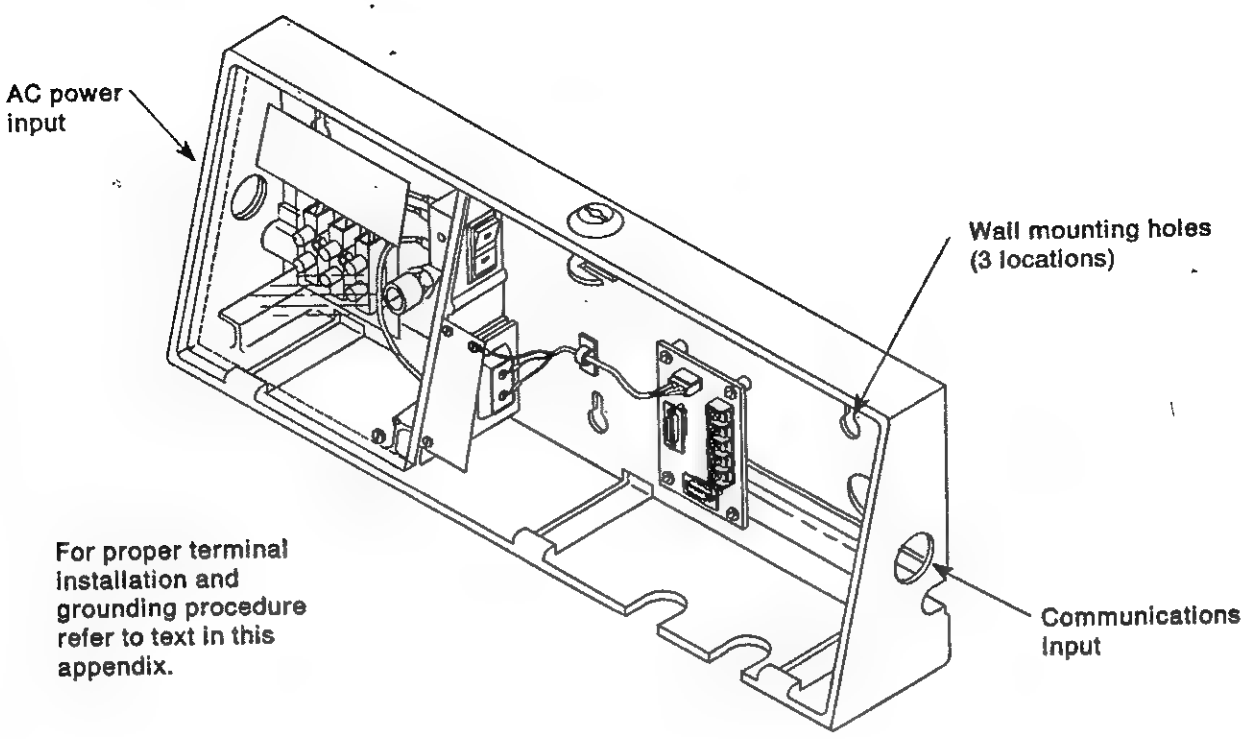
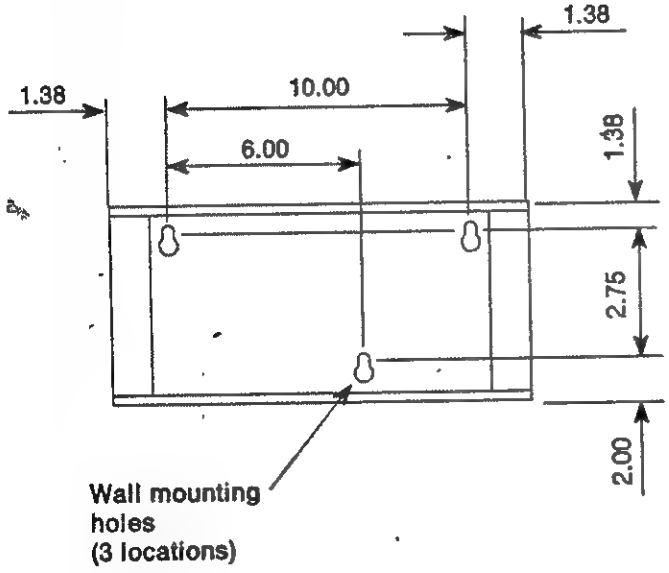


Figure D-1. Unit Dimensions for Desk Mount





Note: The two top mounting holes should be located a minimum of 7.38 inches below the nearest obstruction to allow room for operation of key lock.

Figure D-2. Unit Dimensions for Wall Mount

Installation Checklist

Task, procedure, or checkpoint	Date completed
Identify project leader	
Identify who will install cables and accessories (outside electrical contractor or employees)	
Define the installation wiring (equipment locations and cable routing)	
Obtain management approval for overall installation	
Order terminals and accessories	
Define machine readable media and order (bar code or magnetic badges)	
Define custom overlay for terminal and order (function key layout and graphics)	
Confirm deliveries of all required components	
Install communications and power wiring	
Check out wiring, voltages, and continuity	
Verify safety and code requirements are met	
Complete physical installation of terminals	
Complete electrical installation of terminals	
Select initial communications parameters (terminal addresses)	
Load required programs into the host computer	
Check out terminal operation	

Typical Configurations

A block diagram of a typical 7525 terminal configuration is shown on page D-7.

Field Wiring (Cables and Connectors)

The following tables show methods of wiring, types of cables, and connector types required to connect 7525 Data Collection Terminals to a host computer. The host computer can be an IBM Series/1 with and RS422 feature card installed, an IBM Personal Computer with an RS422 feature card installed, or an IBM 7531/7532 Industrial Computer with an RS422 feature card installed.

Table D-1. Types of RS422 Field Wiring

Cable Description	Mfg. No.	Wire Size
IBM - RS422 cable	5770	22 AWG
Belden - two, twisted-pair, stranded with overall bundle shield	9512 9502	22 AWG 24 AWG
Alpha - two, twisted-pair, stranded with overall bundle shield	5122 5482 5272 5492	22 AWG 22 AWG 24 AWG 24 AWG

Note: Source and part number may be cross referenced to find an equivalent part.

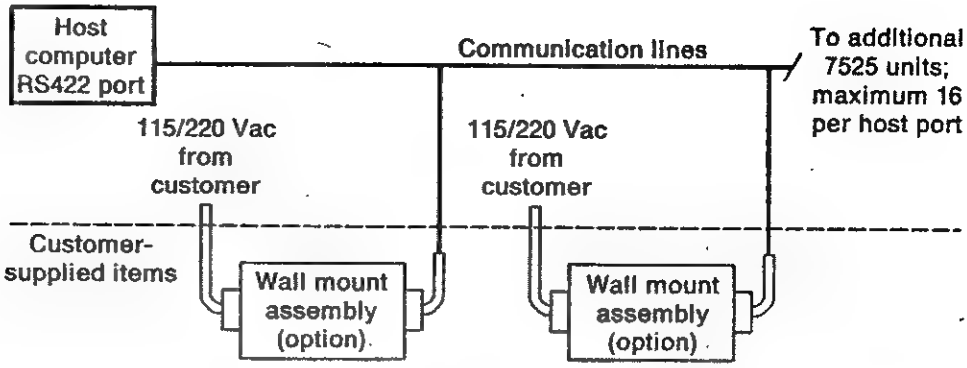
Table D-2. Types of Connectors (text)

Connector Description	Manufacturer	Part Number
9-pin, D-shell, male dataphone connector	TRW CINCH	DE-9P
9-pin, D-shell, female dataphone connector	TRW CINCH	DE-9S
2 x 4-pin Berg connector kit (one connector and 16 pins, sockets)	IBM	6843689
2 x 8-pin Berg connector kit (one connector and 16 pins, sockets)	Berg	5405-005 47712

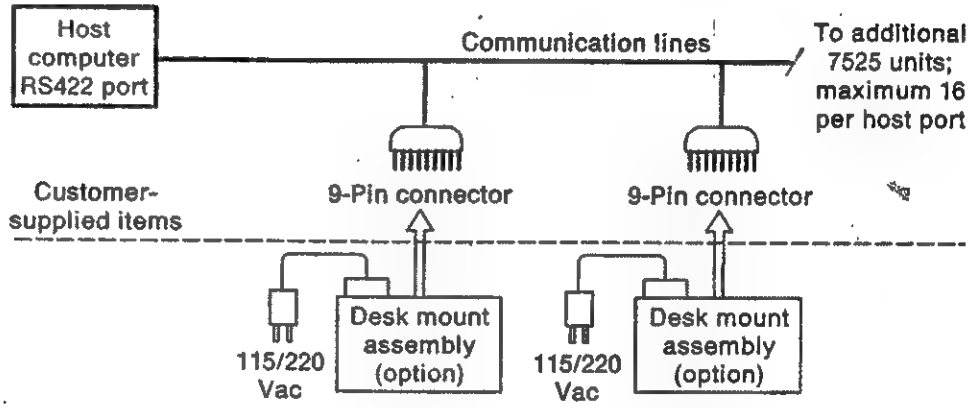
Note: Source and part number may be cross referenced to find an equivalent part.

Typical Cable Block Diagrams

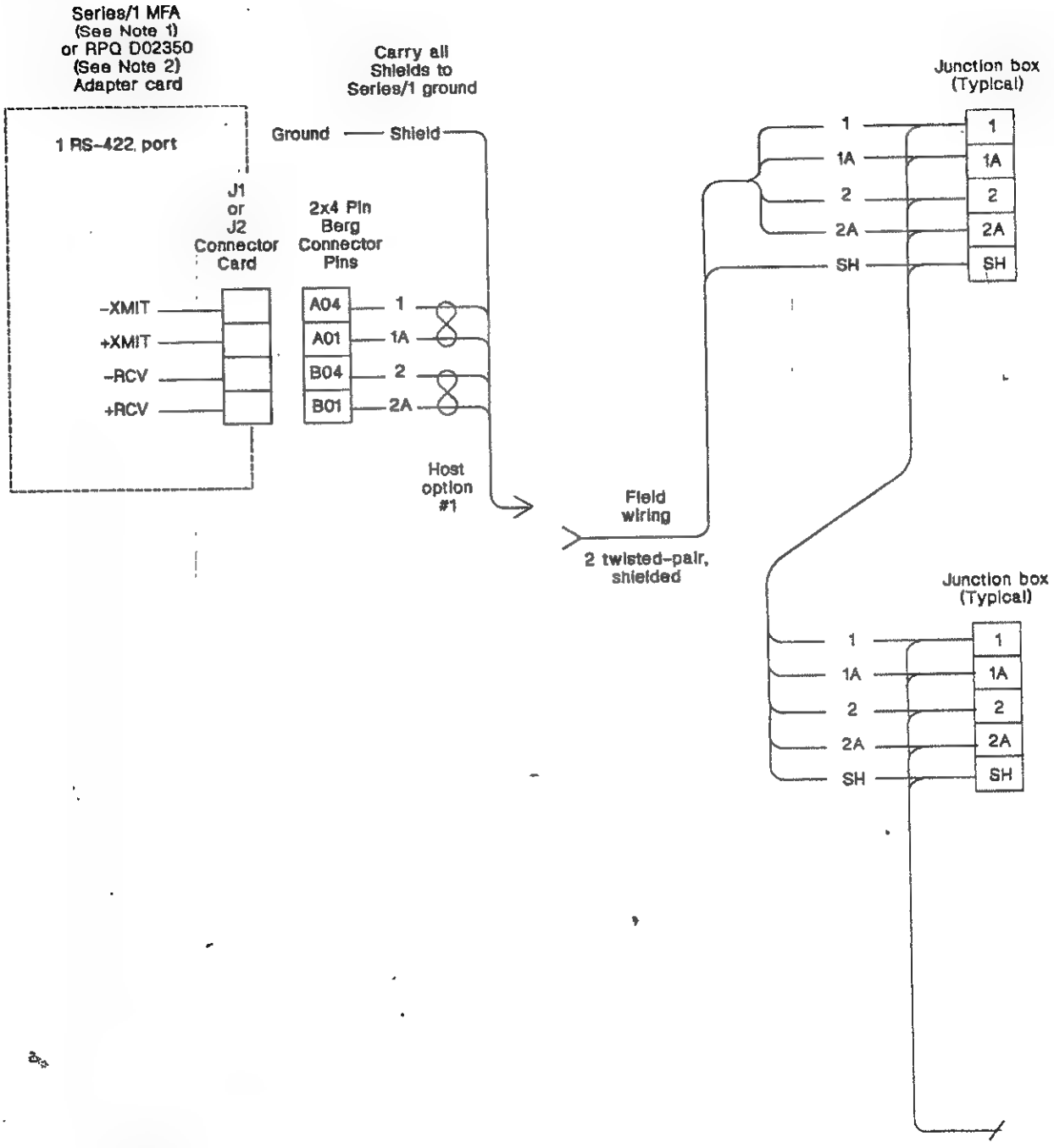
Wall Mount Accessory



Desk Mount Accessory



Typical RS422 Host Communication Wiring Schematic



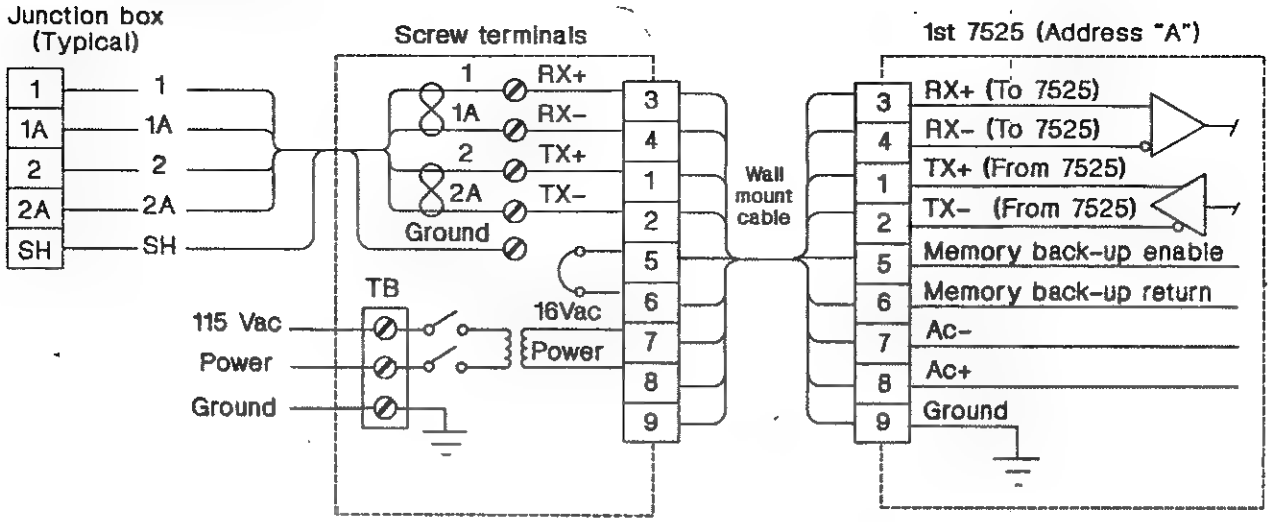
Notes:

1.) Each MFA has 4 RS422 ports.
Adapter Card has 2 X 16 pin Berg connector

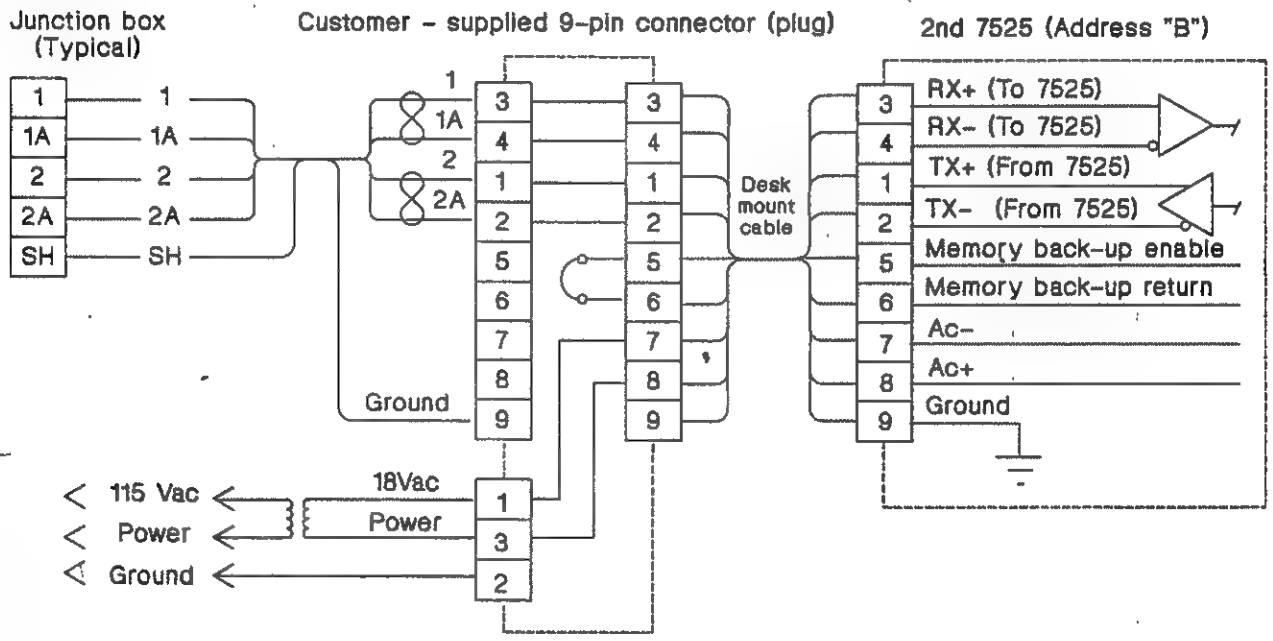
2.) Each RPQ D02350 has 8 RS-422 ports.
Adapter card has 2 X 16 pin Berg connector
J1 and J2 each port is wired to a 2 X 4
Berg connector as shown above. Port/line
selection is made by plugging up to
four 8-pin connectors into each 32 pin connector

Typical RS422 Terminal Wiring Schematics

Wall Mount Accessory



Desk Mount Accessory



Appendix E. A Sample Program

The following Basic program demonstrates communication operations between a 7525 Terminal and a Personal Computer. The program defines four keys on the terminal and downloads the appropriate files to allow these keys to be active. After the download, the program continuously polls the terminal for transactions and displays them when they are received at the PC.

The four keys which are defined are A, B, C and G. The functions of each key are:

Key A	Reads a seven digit badge from the magnetic reader or bar code wand (code 3 of 9).
Key B	Reads a seven digit badge. The day of the week must be typed in from the keyboard and is then verified against the valid days of the week.
Key C	Reads a seven digit badge. The month of the year must be typed in from the keyboard and is then verified against the valid months of the year.
Key G	This is a "hot" key. It sends a time and date transaction to the PC without any further input from the keyboard or badge reader.

The program illustrates the proper transmit and receive protocol for polling and sending commands to the terminal.

After the initial download, the program can be terminated by pressing and holding any active key on the PC keyboard.

The program steps are as follows:

```
200  '
205  '
210  ' Initialize Protocol Variables
215  '
220    STX$ = CHR$(2)  : ACK$ = CHR$(6)  : RS$  = CHR$(30) : DC1$ = CHR$(17)
225    ETX$ = CHR$(3)  : NAK$ = CHR$(21) : CAN$ = CHR$(24) : DC2$ = CHR$(18)
230    ESC$ = CHR$(27) : ADDR$ = "aa"    : KEY OFF
235  '
240  '
245  ' Open Communications Port 1 with 9600 baud rate, no parity, 8 data
250  ' bits, 1 stop bit, suppressing RTS, CTS, DSR and CD.
255  '
260    OPEN "COM1:9600,N,8,1,rs,cs,ds,cd" AS #1
265  '
270  '
275  ' The following lines are the main loop for the program.  Here is
280  ' what they are doing:
285  '
290  '    1) Send a poll to the terminal (The STX, address, and ETX are
300  '       appended to XMIT$ in the transmit routine- at line 3000)
305  '    2) If the terminal returns DC1 then goto the download routine- at
310  '       line 5000
315  '    3) Otherwise if data is returned- then display the data
320  '    4) Otherwise check if a key was pressed to end the program
325  '    5) If no key was pressed goto step 1 above and send another poll
330  '
335      XMIT$ = ""          : GOSUB 2000 : GOSUB 3000   ' Poll terminal
340      IF RETDATA$ = DC1$             THEN GOSUB 5000   ' Do download
```

```
345     IF LEFT$(RETDATA$,1) = STX$  THEN GOSUB 400    ' If data then display
350     A$ = INKEY$ : IF INKEY$ = "" THEN 335
355     PRINT "You pressed a key to stop the program . . ."
360     END
400   '
405   '
410   ' Display Data Received from Terminal
415   '
420     SHOW$ = MID$(RETDATA$,2,LEN(RETDATA$) - 4)
425     PRINT "Received from terminal: ";show$
430     PRINT
435     RETURN
1000  '
1005  ' Transmit and Verify a Command Routine
1010  '
1015  '
1020    GOSUB 2000                                  ' Time delay routine
1025    GOSUB 3000                                  ' Transmit data routine
1030    XMIT$ = "1"                                 ' Get terminal status
1035    GOSUB 2000                                  ' Time delay routine
1040    GOSUB 3000                                  ' Transmit data routine
1045    RETURN
2000  '
2005  '
2010  ' Time Delay Routine
2015  '
2020    NOWSEC = VAL(RIGHT$(TIME$,1))               ' Get current second
2025    NXTSEC = NOWSEC + 1                         ' Figure next second
2030    IF NXTSEC > 9 THEN NXTSEC = NXTSEC - 10     ' Adjust for 10
2035    CURSEC = VAL(RIGHT$(TIME$,1))               ' Get current second
2040    IF CURSEC < NXTSEC THEN 2035                ' If same wait some more
2045    RETURN
3000  '
3005  '
3010  ' Transmit Data Routine
3015  '
3020    TIMEOUT  = 800 : TIMECOUNT = 0              ' Initialize timer vars
3025    TXMITLEN = LEN(XMIT$)                       ' Get length of data
3030    IF TXMITLEN = 0 then 3045  '                ' If poll cmd goto send
3035    CHECK$ = XMIT$ : GOSUB 4000                 ' Go get checksum chars
3040    XMIT$  = XMIT$ + CHK$                       ' Append checksum chars
3043                                                '
3045    PRINT #1, STX$ + ADDR$ + XMIT$ + ETX$       ' Send data to terminal
3047    REM print "Command: "; xmit$                ' (For Diagnostics)
3050    TIMECOUNT = TIMECOUNT + 1                   ' Increment timer
3055    IF TIMECOUNT > TIMEOUT THEN 3065            ' If timed out then break
3060    IF EOF(1) THEN 3050                         ' If no response yet, wait
3063                                                '
3065    RETDATA$ = "" : DATALEN = 0                 ' Initialize indata vars
3070    IF LOC(1) = 0 THEN ? "Timeout" : RETURN     ' If no response, time out
3075    RETCHAR$ = INPUT$(1,#1)                     ' Otherwise get a char
3077    REM print "** ";val(retchar$);"  ";retchar$  ' (For Diagnostics)
3080    RETDATA$ = RETCHAR$                         ' Set return data to input
3081    IF RETCHAR$ = STX$ THEN 3085                ' If data goto get data
3082    RETURN                                      '
3083                                                '
3085    RETCHAR$ = INPUT$(1,#1)                     ' Get next character
3090    REM print "** ";val(retchar$);"  ";retchar$  ' (For Diagnostics)
```

```
3095    RETDATA$ = RETDATA$ + RETCHAR$          ' Build return data
3100    DATALEN  = DATALEN  + 1                 ' Incr data length
3105    IF DATALEN  > 100  THEN RETURN          ' If unreasonable, return
3110    IF RETCHAR$ = ETX$ THEN 3125            ' If end, verify chksum
3115    GOTO 3085                               ' Get more characters
3120                                            '
3125    LENRECV  = LEN(RETDATA$)                ' Get length of input
3130    CHKSMPOS = LENRECV - 2                  ' Find chksum position
3135    LENDATA  = LENRECV - 4                  ' Get length of data
3140    CHECK$   = MID$(RETDATA$,2,LENDATA)     ' Get chars to chksum
3145    GOSUB 4000                              ' Calculate checksum
3150    RECVCHKS$ = MID$(RETDATA$,CHKSMPOS,2)   ' Get checksum received
3155    IF RECVCHKS$ <> CHKS$ THEN  3210        ' if fails goto fail chksm
3160                                            '
3165    PRINT #1, ACK$                            ' Send OK to terminal
3170    IF LEFT$(RETDATA$,1) <> STX$ THEN 3200    ' If not txtion, return
3175    IF VAL(MID$(RETDATA$,2,1)) < 5 THEN 3200  ' If not interactive, return
3180    TXTION$ = RETDATA$                        ' Store txtion for return
3185    XMIT$ = "B1#Transaction Accepted" + RS$   ' Set response for terminal
3190    GOSUB 1000                                ' Transmit response
3195    RETDATA$ = TXTION$                        ' Retrieve txtion for return
3200    RETURN                                    '
3205                                            '
3210    PRINT #1, NAK$                          ' Send FAIL to terminal
3215    RETDATA$ = "" : RETURN                  ' Set data to null
4000  '
4005  '
4010  '
4015  ' Calculate Checksum Characters- gets CHECK$, sets CHKS$
4020  '
4025    CHECKTOT% = 0
4030    FOR CNT% = 1 TO LEN(CHECK$)
4035      CHECKTOT% = CHECKTOT% + ASC(MID$(CHECK$,CNT%,1)) AND 255
4040    NEXT
4045    CHKS$ = RIGHT$(HEX$(CHECKTOT% OR 256), 2)
4050    RETURN
5000  '
5005  '
5010  '
5015  ' Download to the Terminal
5020  '
5025  '   Initialize the files- specifying their size and types.  The
5030  '   format is   9Bfllllldt   where f is the file to be initialized,
5035  '   llll is the number of bytes to reserve, d is the type of text
5040  '   to be stored in the file and t is the type of text file.
5045  '
5047    GOSUB 7000 : PRINT "Download in progress . . . wait 2 minutes."
5050    XMIT$ = "9B0100000" + RS$ : GOSUB 1000   ' Init File 0
5055    XMIT$ = "9B1010000" + RS$ : GOSUB 1000   ' Init File 1
5060    XMIT$ = "9B2010010" + RS$ : GOSUB 1000   ' Init File 2
5065    XMIT$ = "9B3020010" + RS$ : GOSUB 1000   ' Init File 3
5070    XMIT$ = "9B4001000" + RS$ : GOSUB 1000   ' Init File 4
5075    XMIT$ = "9B5001000" + RS$ : GOSUB 1000   ' Init File 5
5080    XMIT$ = "9B6001000" + RS$ : GOSUB 1000   ' Init File 6
5085    XMIT$ = "9B7001000" + RS$ : GOSUB 1000   ' Init File 7
5090    XMIT$ = "9B8050010" + RS$ : GOSUB 1000   ' Init File 8
5095    XMIT$ = "9B9000011" + RS$ : GOSUB 1000   ' Init File 9
5100  '
```

```
5105  '
5110  ' Initialize File 0 Record 0 - Terminal Characteristics
5115  '
5120  '  Byte   Meaning                 Byte   Meaning
5125  '  ----   -------                 ----   -------
5130  '    0    Keypad Type              12    File # - Fast Clck Valdtn
5135  '    1    Badge Rdr Type           13    Type   - Fast Clcl Valdtn
5140  '    2    Buffered/Interactive     14 \  Length of Fast Clocking /
5145  '    3    Time Display Format      15 /   Reprogramming Badge
5150  '    4    Idle Display Format      16 \  Duration of Host Commun-
5155  '    5    Hourly Time Txtion?      17 /   ication Timeout
5160  '    6    Format for Time Stamp    18    Duration Status Msg Display
5165  '    7    Error Logging Mode       19 \  Duration of time between
5170  '    8    # Beeps  - Good Txtion   20 /   operator actions
5175  '    9    Duration - Good Txtion   21    Add Sequence # to Txtion?
5180  '   10    # Beeps  - Bad Txtion    22    Accept Reprogrmmng Badge?
5185  '   11    Duration - Bad Txtion
5190  '
5195    XMIT$ = "9C0140010301122010750620l1" + RS$  : GOSUB 1000   ' Rec 0
5200  '
5205  '
5210  ' Initialize the Key Definitions in File 0
5215  '
5220  '  Key A: Reads badge of length 7 from reader
5225  '  Key B: Reads badge and day of week from keypad and verifies it
5230  '  Key C: Reads badge and month of the year and verifies it
5235  '  Key D - Key F:   Not defined
5240  '  Key G: Hot key- sends time and date stamp transaction
5245  '  Key G - Key F28: Not defined
5250  '
5255    XMIT$ = "9C0:010:10700" + RS$                          : GOSUB 1000   ' Key A
5260    XMIT$ = "9C0:010:10700:011:309211311" + RS$ : GOSUB 1000   ' Key B
5265    XMIT$ = "9C0:010:10700:012:309311311" + RS$ : GOSUB 1000   ' Key C
5270    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key D
5275    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key E
5280    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key F
5285    XMIT$ = "9C0;0" + RS$                                  : GOSUB 1000   ' Key G
5290    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key H
5295    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key I
5300    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key J
5305    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key K
5310    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key L
5315    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key M
5320    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key N
5325    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key O
5330    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key P
5335    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key Q
5340    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key R
5345    XMIT$ = "9C0:9" + RS$                                  : GOSUB I000   ' Key S
5350    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key T
5355    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key U
5360    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key V
5365    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key W
5370    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key X
5375    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key Y
5380    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key Z
5385    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key F27
5390    XMIT$ = "9C0:9" + RS$                                  : GOSUB 1000   ' Key F28
```

```
5395  '
5400  '
5405  ' Initialize File 1 for no fast clocking
5410  '
5415    XMIT$ = "9C1:"  + RS$          : GOSUB 1000
5420  '
5425  '
5430  ' Initialize File 2- contains valid responses for days of
5435  '   the week.  Note all records are the same length: 9.
5440  '
5445    XMIT$ = "9C2SATURDAY " + RS$  : GOSUB 1000
5450    XMIT$ = "9C2SUNDAY   " + RS$  : GOSUB 1000
5455    XMIT$ = "9C2MONDAY   " + RS$  : GOSUB 1000
5460    XMIT$ = "9C2TUESDAY  " + RS$  : GOSUB 1000
5465    XMIT$ = "9C2WEDNESDAY" + RS$  : GOSUB 1000
5470    XMIT$ = "9C2THURSDAY " + RS$  : GOSUB 1000
5475    XMIT$ = "9C2FRIDAY   " + RS$  : GOSUB 1000
5480  '
5485  '
5490  ' Initialize File 3- contains valid responses for months of
5495  '   the year.  Note all records are the same length: 9.
5500  '
5505    XMIT$ = "9C3JANUARY  " + RS$  : GOSUB 1000
5510    XMIT$ = "9C3FEBRUARY " + RS$  : GOSUB 1000
5515    XMIT$ = "9C3MARCH    " + RS$  : GOSUB 1000
5520    XMIT$ = "9C3APRIL    " + RS$  : GOSUB 1000
5525    XMIT$ = "9C3MAY      " + RS$  : GOSUB 1000
5530    XMIT$ = "9C3JUNE     " + RS$  : GOSUB 1000
5535    XMIT$ = "9C3JULY     " + RS$  : GOSUB 1000
5540    XMIT$ = "9C3AUGUST   " + RS$  : GOSUB 1000
5545    XMIT$ = "9C3SEPTEMBER" + RS$  : GOSUB 1000
5550    XMIT$ = "9C3OCTOBER  " + RS$  : GOSUB 1000
5555    XMIT$ = "9C3NOVEMBER " + RS$  : GOSUB 1000
5560    XMIT$ = "9C3DECEMBER " + RS$  : GOSUB 1000
5565  '
5570  '
5575  ' Initialize File 8- message file.  Messages 0 - 9 are reserved
5580  '   for specific functions.  Messages 10 - 12 are user defined.
5585  '   Note File 8 records can be of variable length.
5590  '
5595    XMIT$ = "9C8Press a key, any key . . ."          + RS$ : GOSUB 1000
5600    XMIT$ = "9C8Your time is up "                    + RS$ : GOSUB 1000
5605    XMIT$ = "9C8Length of input is incorrect"        + RS$ : GOSUB 1000
5610    XMIT$ = "9C8Input data is invalid"               + RS$ : GOSUB 1000
5615    XMIT$ = "9C8Invalid Operation- press key first" + RS$ : GOSUB 1000
5620    XMIT$ = "9C8Fast Clocking now active . . . "     + RS$ : GOSUB 1000
5625    XMIT$ = "9C8Please wait one moment . . ."        + RS$ : GOSUB 1000
5630    XMIT$ = "9C8Transaction accepted, thank you"     + RS$ : GOSUB 1000
5635    XMIT$ = "9C8Successful transaction"              + RS$ : GOSUB 1000
5640    XMIT$ = "9C8Press another active function key"  + RS$ : GOSUB 1000
5645    XMIT$ = "9C8Enter your badge please . . ."       + RS$ : GOSUB 1000
5650    XMIT$ = "9C8Enter the day of the week . . .   " + RS$ : GOSUB 1000
5655    XMIT$ = "9C8Enter the month of the year . . . " + RS$ : GOSUB 1000
5660  '
5665  '
5670  '  Specify File 9 to buffer transactions- send 1 txtion per poll
5675  '
5680    XMIT$ = "8B91" + RS$      : GOSUB 1000
```

```
5685  '
5690  '
5695  ' Get the time and date from PC and send it to the terminal
5700  '
5705    GOSUB 6000 : XMIT$ = "4"  + DATTIM$ + RS$ : GOSUB 1000
5710  '
5715  '
5720  ' Put the terminal in service- ready to collect transactions
5730  '
5740    XMIT$ = "A1" + RS$      : GOSUB 1000
5745    GOSUB 7000 : PRINT "Download complete . . . polling terminal" : PRINT
5750    RETURN
6000  '
6005  '
6010  ' Get the Current Date and Time from the PC
6015  '
6020    NOWTIME$ = LEFT$(TIME$,2)  + MID$(TIME$,4,2) + RIGHT$(TIME$,2)
6025    NOWDATE$ = RIGHT$(DATE$,2) + LEFT$(DATE$,2)  + MID$(DATE$,4,2)
6030    DATTIM$  = NOWDATE$ + NOWTIME$
6035    RETURN
7000  '
7005  '
7010  ' Clear screen and move cursor to the middle row on the screen
7015  '
7020    CLS : FOR ROW = 1 TO 12 : PRINT : NEXT
7025    RETURN
```

G

H

I

J

K

L

M

N

O

P

Numerics